#10821687-2

Access DB# 151055

SEARCH REQUEST FORM

Scientific and Technical Information Center

Requester's Full Name: GWEN LIANG Examiner #: 79180 Date: 4-18-05
Art Unit: 2162 Phone Number ~~30~~ x 24038 Serial Number: ~~10/821,539~~ 10/821687
Mail Box and Bldg/Room Location: RND 3B11 Results Format Preferred (circle): PAPER DISK E-MAIL

If more than one search is submitted, please prioritize searches in order of need.

****

Please provide a detailed statement of the search topic, and describe as specifically as possible the subject matter to be searched. Include the elected species or structures, keywords, synonyms, acronyms, and registry numbers, and combine with the concept or utility of the invention. Define any terms that may have a special meaning. Give examples or relevant citations, authors, etc, if known. Please attach a copy of the cover sheet, pertinent claims, and abstract.

Title of Invention: System and Method for Fragment-Based Serialization

Inventors (please provide full names): TEREK, Soneri; KALHAN, Ajay; PONNEKANTI, Nagavamsi; RANGARAJAN, Srikumar; ZWILLING, Michael J.

Earliest Priority Filing Date: 4/9/2004

***For Sequence Searches Only* Please include all pertinent information (parent, child, divisional, or issued patent numbers) along with the appropriate serial number.**

Background = the unaddressed need in Serialization. (See CON pages 1-5)

- Concept of invention: (See CON page 3)

Claim 1 (independent claim to focus)
2-13 (dependent claims for overall understanding)
(See CLM pages)

* Assignee: Micro soft Corporation

* keyword: serialization

* drawings: (show Serialization examples, inclu. prior art) (DRAWS pages)

RECEIVED
APR 19 2005
BY:

****

STAFF USE ONLY

Searcher: EMORY DAMRON
Searcher Phone #: 2-3520
Searcher Location: RND 4B19
Date Searcher Picked Up: 5/11/05 3PM
Date Completed: 5/13/05 930A
Searcher Prep & Review Time: 270 M
Clerical Prep Time: 0
Online Time: 270 M

Type of Search

NA Sequence (#) _____
AA Sequence (#) _____
Structure (#) _____
Bibliographic X
Litigation _____
Fulltext X
Patent Family _____
Other _____

Vendors and cost where applicable

STN _____
Dialog X [redacted] 130366
Questel/Orbit _____
Dr.Link _____
Lexis/Nexis _____
Sequence Systems _____
WWW/Internet X
Other (specify) _____

PTO-1590 (8-01)

```
Set     Items   Description
S1       1329   SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR -
            CLRSERIALIZ? OR CLRSERIALIS?
S2      10850   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
            ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3     704163   FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT?
             OR INCHOAT?
S4     564781   SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM
            OR BYTE?)
S5      82023   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
            OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV?)
S6       1348   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7      14345   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
            DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
            M)
S8       1769   DATAOBJECT? OR DATA()OBJECT?
S9         76   DATAMEMBER? OR DATA()MEMBER?
S10     16876   (RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11     56876   (LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PRED-
            ICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED
             OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12   1587053   ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR F-
            LANK? OR BESIDE OR CLOSE
S13     94519   INSTANTIAT? OR UPDAT? OR UP()(DATE? OR DATING?)
S14   1201924   IC=G06F?
S15      2524   S1:S2 AND S3:S9
S16       788   S15 AND S8
S17        31   S16 AND S8 AND (S7 OR S9)
S18         0   S16 AND S9
S19      1642   S15 AND S3:S6
S20       239   S19 AND S1 AND S3:S6
S21       149   S20 AND S10:S14
S22       126   S15 AND S1:S2(5N)S3
S23         8   S22 AND S21
S24       267   S21:S22
S25       164   S24 AND S4:S9
S26        17   S24 AND S11
S27       145   S25 AND S14
S28       139   (S24 OR S27) AND S4
S29         7   (S24 OR S27) AND S5:S6
S30       267   S24 OR S27
S31        18   S30 AND S8:S9
S32       349   S20 OR S21 OR S22 OR S24 OR S25 OR S27 OR S28 OR S30
S33       123   S32 AND (S1 OR S3)/TI
S34         7   S32 AND (S1 AND S3)/TI
S35        17   S32 AND S11
S36        80   S17 OR S23 OR S26 OR S29 OR S31 OR S34:S35
S37        30   S36 AND S33
S38        80   S36:S37
S39       287   PR=2005
S40        80   S38 NOT S39
S41        80   IDPAT (sorted in duplicate/non-duplicate order)
? show files
File 347:JAPIO Nov 1976-2005/Jan(Updated 050506)
        (c) 2005 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200529
        (c) 2005  Thomson Derwent
?
```

41/3,K/1 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

016661961 **Image available**
WPI Acc No: 2004-820680/200481
XRPX Acc No: N04-647922
XML serialized **image data transferring method for relational database system, involves sending message including** payload **with** serialized **data for construct and** type **field with data of selected format between components of system**
Patent Assignee: ORACLE INT CORP (ORAC-N)
Inventor: CHANDRASEKAR S; JALALI N; KRISHNAPRASAD M; MANIKUTTY A; MURTHY R; WARNER J
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20040220946	A1	20041104	US 2003428393	A	20030501	200481 B

Priority Applications (No Type Date): US 2003428393 A 20030501
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20040220946	A1		20	G06F-007/00	

XML serialized **image data transferring method for relational database system, involves sending message including** payload **with** serialized **data for construct and** type **field with data of selected format between components of system**

RELATED DOC. BENEATH

Abstract (Basic):
... The method involves selecting a format from different XML **serialization** formats that represent data for XML constructs (144a, 144b) as a series of data units. A message including a **payload** with **serialized** data for a particular construct and a **type** field with data of the selected format is generated. The message is sent from a...
... An INDEPENDENT CLAIM is also included for computer-readable medium carrying **sequences** of instructions for transferring a **serialized** image of data for an XML construct...

...Used for transferring XML **serialized** image data between components of a relational database system...
International Patent Class (Main): **G06F-007/00**

41/3,K/14 (Item 14 from file: 350)
DIALOG(R)File 350:Derwent WPIX
(c) 2005 Thomson Derwent. All rts. reserv.

015685593 **Image available**
WPI Acc No: 2003-747782/200370
XRPX Acc No: N03-599463
Data object processing **method for** fragment **caching in Internet, involves determining caching of** fragment **by computing device that has** fragment **-supporting cache management unit, after receiving response for** fragment **request**
Patent Assignee: INT BUSINESS MACHINES CORP (IBMC)
Inventor: AGARWALLA R S; CHALLENGER J R H; COPELAND G P; IYENGAR A K; MEDURI S
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030188016	A1	20031002	US 200134770	A	20011219	200370	B

Priority Applications (No Type Date): US 200134770 A 20011219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030188016	A1		68	G06F-015/173	

Data object processing **method for** fragment **caching in Internet, involves determining caching of** fragment **by computing device that has** fragment **-supporting cache management unit, after receiving response for** fragment **request**

Abstract (Basic):
... A computing device receives request message with source identifier, for **fragment** and determines whether the request message is processed by another computing device. The **fragment** caching by the another device having the **fragment** - supporting cache management unit is determined after the response message is received.
... 2) a computer program product for **data object processing**
...

...For **object data processing** by **fragment** caching, in Internet...

...Provides distributed **fragment** caching mechanism by the cache management unit, thus reduces cache size provided by the **fragment** compression. Hence cost is reduced and performance is improved...

...The figure shows the block diagram of typical web page composed of **fragments** .
...
...dynamic content **fragments** (200
...Title Terms: **FRAGMENT** ;
International Patent Class (Main): **G06F-015/173**

41/3,K/20 (Item 20 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015599596 **Image available**
WPI Acc No: 2003-661751/200362
XRPX Acc No: N03-528027
Segmented **distributed memory module cache apparatus has command** sequencer **and** serializer **unit coupled to array of** tag address **storage** locations , **in which each** tag address **storage** location **corresponds to cache line divided into two** segments
Patent Assignee: DAVID H S (DAVI-I); INTEL CORP (ITLC)
Inventor: DAVID H S
Number of Countries: 001 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
US 20030126363	A1	20030703	US 200139612	A	20011231	200362	B
US 6865646	B2	20050308	US 200139612	A	20011231	200518	

Priority Applications (No Type Date): US 200139612 A 20011231
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20030126363	A1		11	G06F-012/08	
US 6865646	B2			G06F-012/00	

Segmented **distributed memory module cache apparatus has command** sequencer **and** serializer **unit coupled to array of** tag address **storage** locations , **in which each** tag address **storage** location **corresponds to cache line divided into two** segments

Abstract (Basic):
... The apparatus has a command **sequencer** and **serializer** unit coupled to an array of **tag address** storage **locations** to control a data cache **associated** with a memory module (220,230,240,250). Each **tag address** storage **location** corresponds to a cache line divided into two **segments** .
... An INDEPENDENT CLAIM is included for the **segmented** distributed memory module cache system...

... **Segmented** distributed memory module cache apparatus...

...Allows entire cache to accessed without doubling the amount of **tag address** storage **locations** .

Title Terms: **SEGMENT** ;
International Patent Class (Main): **G06F-012/00** ...

... **G06F-012/08**

41/3,K/25 (Item 25 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015087487 **Image available**
WPI Acc No: 2003-148005/200314
Related WPI Acc No: 2003-103028; 2003-138827; 2003-199373; 2003-199770; 2003-209556; 2003-266142; 2003-391839; 2003-391840; 2003-777106; 2003-800949
XRPX Acc No: N03-116938

Data object **association implementation method in distributed computing environment, involves forming pair of association** fragments **comprising information relevant to accessing respective** data objects

Patent Assignee: GREENE W S (GREE-I); ROBERTSON J A (ROBE-I)
Inventor: GREENE W S; ROBERTSON J A
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
US 20020165727	A1	20021107	US 2000206564	P	20000522	200314 B
			US 2001863456	A	20010522	

Priority Applications (No Type Date): US 2000206564 P 20000522; US 2001863456 A 20010522
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
US 20020165727	A1		99	G06F-007/00	Provisional application US 2000206564

Data object **association implementation method in distributed computing environment, involves forming pair of association** fragments **comprising information relevant to accessing respective** data objects

RELATED DOC. BENEATH

Abstract (Basic):
... The method involves forming a pair of association **fragments** comprising information relevant to accessing respective **data objects** that are maintained in separate **data stores** . The association **fragments** cooperate to effectively form an association between the **data objects** .
... For implementing an association among **data objects** in a distributed computing environment in large service company such as global telecommunications enterprise...
...Title Terms: **FRAGMENT** ;
International Patent Class (Main): **G06F-007/00**

41/3,K/36 (Item 36 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014269410 **Image available**
WPI Acc No: 2002-090108/200212
XRPX Acc No: N02-066357
Non-hierarchical file sub-system stores byte **portions of** data objects **contiguously in** segment **with round-robin selection of new stored objects**
Patent Assignee: INFOLIBRIA INC (INFO-N)
Inventor: MORRIS R J; RABII F
Number of Countries: 096 Number of Patents: 004
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200193106	A2	20011206	WO 2001US17230	A	20010525	200212 B
US 20020032691	A1	20020314	US 2000207995	P	20000526	200222
			US 2001866383	A	20010525	
AU 200165075	A	20011211	AU 200165075	A	20010525	200225
EP 1358575	A2	20031105	EP 2001939572	A	20010525	200377
			WO 2001US17230	A	20010525	

Priority Applications (No Type Date): US 2001866383 A 20010525; US 2000207995 P 20000526
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200193106	A2	E	32	G06F-017/30	

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT TZ UA UG UZ VN YU ZA ZW
Designated States (Regional): AT BE CH CY DE DK EA ES FI FR GB GH GM GR IE IT KE LS LU MC MW MZ NL OA PT SD SE SL SZ TR TZ UG ZW

US 20020032691	A1		G06F-012/00	Provisional application US 2000207995
AU 200165075	A		G06F-017/30	Based on patent WO 200193106
EP 1358575	A2	E	G06F-017/30	Based on patent WO 200193106

Designated States (Regional): DE FR GB
Non-hierarchical file sub-system stores byte **portions of** data objects **contiguously in** segment **with round-robin selection of new stored objects**

Abstract (Basic):
... disk drive allocated by the server with multiple object data partitions containing multiple fixed-length **segments** , plus a meta disk partition for **storing** sub-system meta **data** and **object** meta **data** . An **object** directory comprises an array of directory blocks each with pointers to a particular disk object space within a **segment** , data being retrieved using a hash value of a hierarchical specifier (URL) for the **data object** . The first hash portion is an index for selecting directory blocks and the second selects...
... Data buffers are allocated to the file sub-system to receive and return **data objects sequentially** in response to requests for objects using a hash value representing a URL. The retrieved **data object** has a header comprising the size and URL of the **data object** and a trailer comprising a two-part hash value representing the **data object** . When a **segment** is full a **data object** overwrites the oldest **data object** in the **segment** .
...

...Sub-system is for **storing** **data** **objects** and is a disk file structure
...Title Terms: **SEGMENT** ;
International Patent Class (Main): **G06F-012/00** ...

... **G06F-017/30**

41/3,K/62 (Item 62 from file: 350)
DIALOG(R)File 350:Derwent WPIX

009467409
WPI Acc No: 1993-160948/199320
XRPX Acc No: N93-123524

Logical mapping of data objects **using data spaces - uses uniquely identified concatenated sub-data space to place and access** data objects **of various sizes within simulated contiguous data space**

Patent Assignee: IBM CANADA LTD (IBMC); INT BUSINESS MACHINES CORP (IBMC)
Inventor: FECTEAU J G; KLIGERMAN E; KOLLAR L
Number of Countries: 005 Number of Patents: 013
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 542483	A1	19930519	EP 92310160	A	19921105	199320 B
CA 2055295	A	19930513	CA 2055295	A	19911112	199330
US 5561778	A	19961001	US 92975245	A	19921112	199645
			US 95442770	A	19950517	
US 5579499	A	19961126	US 92975245	A	19921112	199702
			US 95442770	A	19950517	
			US 95443397	A	19950517	
			US 95468987	A	19950606	
US 5594881	A	19970114	US 92975245	A	19921112	199709
			US 95443397	A	19950517	
US 5652873	A	19970729	US 92975245	A	19921112	199736
			US 95443371	A	19950517	
US 5664160	A	19970902	US 92975245	A	19921112	199741
			US 95443371	A	19950517	
			US 95443397	A	19950517	
			US 95468095	A	19950606	
US 5687343	A	19971111	US 92975245	A	19921112	199751
			US 95443397	A	19950517	
			US 95468771	A	19950606	
CA 2285089	A1	19930513	CA 2055295	A	19911112	200016
			CA 2285089	A	19911112	
CA 2285096	A1	19930513	CA 2055295	A	19911112	200016
			CA 2285096	A	19911112	
CA 2285089	C	20000509	CA 2055295	A	19911112	200037
			CA 2285089	A	19911112	
CA 2285096	C	20000509	CA 2055295	A	19911112	200037
			CA 2285096	A	19911112	
CA 2055295	C	20000523	CA 2055295	A	19911112	200039

Priority Applications (No Type Date): CA 2055295 A 19911112; CA 2285089 A 19911112; CA 2285096 A 19911112
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 542483	A1	E	14	G06F-012/10	
Designated States (Regional): DE FR GB					
CA 2055295	A			G06F-012/10	
US 5561778	A		25	G06F-012/08	Div ex application US 92975245
US 5579499	A		25	G06F-012/08	Cont of application US 92975245
					Div ex application US 95442770
					Div ex application US 95443397
US 5594881	A		25	G06F-012/08	Cont of application US 92975245
US 5652873	A		23	G06F-012/06	Div ex application US 92975245
US 5664160	A		22	G06F-012/08	Cont of application US 92975245
					Div ex application US 95443371
					Div ex application US 95443397

US 5687343	A	25	G06F-012/08	Cont of application US 92975245 Div ex application US 95443397
CA 2285089	A1	E	G06F-012/08	Div ex application CA 2055295
CA 2285096	A1	E	G06F-012/08	Div ex application CA 2055295
CA 2285089	C	E	G06F-012/08	Div ex application CA 2055295
CA 2285096	C	E	G06F-012/08	Div ex application CA 2055295
CA 2055295	C	E	G06F-012/10	

Logical mapping of data objects **using data spaces...**

...uses uniquely identified concatenated sub-data space to place and access data objects **of various sizes within simulated contiguous data space**

...Abstract (Basic): USE/ADVANTAGE - Mapping of **data objects** from a simulated contiguous data space to the memory of a computer system. Allows the...

...Abstract (Equivalent): computer readable program code means embodied in said medium for modifying a page of a **data object** contained in a **database** and **stored** in one or more database **storage** disks, wherein the **database** comprises one or more **database data objects** , wherein the **database** is accessed via a contiguous data space representation, the contiguous data space being represented by...

...addressable by a computer operating system, each sub-data space comprising a plurality of data **segments** , each data **segment** comprising a plurality of pages, wherein the pages in the contiguous data space representation are...

...of addressable storage locations, wherein the contiguous data space, the sub-data spaces, the data **segments** , and the pages are addressable by a database management system, wherein the address of a...

...first computer readable program code means for mapping a **data segment** of the **database data object** from a **database storage** disk to a **data segment** of the sub-data space, wherein said data **segment** from said database storage disk contains the page to be modified... medium for simulating a database in a contiguous data space in computer memory, wherein the **database** comprises one or more **data objects** of variable size and is **stored** in one or more database storage disks, said computer program product having...

...number identifier to record a starting page number in the contiguous data space for each **data object** in the database, wherein said starting page number indicates a page number at which a **data object** is placed in the contiguous data space...

...fourth computer readable program code means for enabling said **processor** to determine whether a **data object** has been added to the contiguous **data** space, when said **data object** is referenced...

...code means for enabling said processor to set said starting page number identifier for said **data object** equal to said next available page identifier if said fourth computer readable program code means determines that said **data object** has not been added to the contiguous data space; and...

...enabling said processor to increment said next available page identifier by the size of said **data object** , wherein said size indicates a number of pages allocated to said **data object** .

...system to simulate a database in a contiguous data space in computer memory, wherein the **database** comprises one or more **data objects** of variable size and is **stored** in one or more database

storage disks, comprising...

...number identifier to record a starting page number in the contiguous data space for each **data object** in the database, wherein said starting page number indicates a page number at which a **data object** is placed in the contiguous data space...

...4) determining whether a **data object** has been added to the contiguous data space, when said **data object** is referenced...

...5) setting said starting page number identifier for said **data object** equal to said next available page identifier if it is determined in step (4) that said **data object** has not been added to the contiguous data space; and...

...6) incrementing said next available page identifier by the size of said **data object** , wherein said size indicates a number of pages allocated to said **data object** .

...A method for modifying a page of a **data object** contained in a **database** and **stored** in one or more database **storage** disks, wherein the **database** comprises one or more **database data objects** , wherein the **database** is accessed via a contiguous data space representation, the contiguous data space being represented by...

...addressable by a computer operating system, each sub-data space comprising a plurality of data **segments** , each data **segment** comprising a plurality of pages, wherein the pages in the contiguous data space representation are...

...of addressable storage locations, wherein the contiguous data space, the sub-data spaces, the data **segments** , and the pages are addressable by a database management system, wherein the address of a...

...1) mapping a **data ' segment** of the **data object** from a **database storage** disk to a **data segment** of the sub-data space, wherein said data **segment** from said database storage disk contains the page to be modified...program code means embodied in said medium for mapping on demand a page of a **data object** contained in a **database** and **stored** in one or more database **storage** disks, wherein the **database** comprises one or more **database data objects** , wherein the **database** is accessed via a contiguous data space representation, the contiguous data space being represented by...

...addressable by a computer operating system, each sub-data space comprising a plurality of data **segments** , each data **segment** comprising a plurality of pages, wherein the pages of the contiguous data space representation are...

...of addressable storage locations, wherein the contiguous data space, the sub-data spaces, the data **segments** , and the pages are addressable by the database management system, wherein the address of a page to be mapped has been determined to be placed in a data **segment** of a sub-data space, said computer program product having...

...third computer readable program code means for creating a **segment** bit map for the sub-data space if said first computer readable program code means...

...the sub-data space has not been created in the contiguous data space, wherein said **segment** bit map comprises a plurality of bits, each bit representing a data **segment** contained in the sub-data space and

indicating whether said data **segment** has been mapped to the database storage disk...

...fourth computer readable program code means for determining if a bit in said **segment** bit map is equal to a predetermined value, wherein said bit corresponds to the data **segment** of the sub-data space containing the page to be mapped, and said predetermined value indicates that the data **segment** corresponding to said bit has been mapped to the database storage disk...

...fifth computer readable program code means for mapping a **data segment** of the **data object** from the **database storage** disk to the **data segment** of the sub-data space, wherein said data **segment** from the database storage disk contains the page to be mapped, and said mapping occurs if said fourth computer readable program code means determines that said bit in said **segment** bit map does not equal said predetermined value; and...

...sixth computer readable program code means for setting said bit in said **segment** bit map to said predetermined value, wherein said setting occurs when said fifth computer readable...A method for mapping on demand a page of a **data object** contained in a **database** and **stored** in one or more database **storage** disks, wherein the **database** comprises one or more **database data objects** , wherein the **database** is accessed via a contiguous data space representation, the contiguous data space being represented by...

...addressable by a computer operating system, each sub-data space comprising a plurality of data **segments** , each data **segment** comprising a plurality of pages, wherein the pages in the contiguous data space representation are...

...of addressable storage locations, wherein the contiguous data space, the sub-data spaces, the data **segments** , and the pages are addressable by a database management system, wherein the address of a page to be mapped has been determined to be placed in a data **segment** of a sub-data space, the method comprising the steps of...

...3) creating a **segment** bit map for the sub-data space if step (1) determines that the sub-data space has not been created in the contiguous data space, wherein said **segment** bit map comprises a plurality of bits, each bit representing a data **segment** contained in the sub-data space and indicating whether said data **segment** has been mapped to the database storage disk...

...4) determining if a bit in said **segment** bit map is equal to a predetermined value, wherein said bit corresponds to the data **segment** of the sub-data space containing the page to be mapped, and said predetermined value indicates that the data **segment** corresponding to said bit has been mapped to the database storage disk...

...5) mapping a **data segment** of the **data object** from the **database storage** disk to the **data segment** of the sub-data space, wherein said data **segment** from the database storage disk contains the page to be mapped, and said mapping occurs if step (4) determines that said bit in said **segment** bit map does not equal said predetermined value; and ...

...6) setting said bit in said **segment** bit map to said predetermined value, wherein said setting occurs when step (5) is performed...

International Patent Class (Main): **G06F-012/06** ...

... **G06F-012/08** ...

... **G06F-012/10**
International Patent Class (Additional): **G06F-012/02** ...

... **G06F-017/30**

```
Set     Items   Description
S1      10859   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
             ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S2      23655   (SEQUENT? OR SEQUENC? OR SERIAL?)(5N)(STORE? OR STORING OR
             STORAG?)
S3      51157   (FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCA-
             T? OR INCHOAT? OR SNIPPET? OR BYTE? OR LENGTH? OR PRIMITIV?)(-
             7N)(FIXED? OR STATIC? OR VARIAB? OR COMPLEX?)
S4      82064   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
             OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S5       1350   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S6      14367   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
             DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
             M)
S7       1770   DATAOBJECT? OR DATA()OBJECT?
S8         76   DATAMEMBER? OR DATA()MEMBER?
S9    1203146   IC=G06F?
S10        20   S1:S2 AND S3 AND S4:S8
S11        16   S10 AND S9
S12        20   S10:S11
? show files
File 347:JAPIO Nov 1976-2005/Jan(Updated 050506)
         (c) 2005 JPO & JAPIO
File 350:Derwent WPIX 1963-2005/UD,UM &UP=200530
         (c) 2005  Thomson Derwent
?
```

PAT LIT

BIBLIOG FILES

DIFFERENT STRATEGY

12/3,K/2 (Item 1 from file: 350)
DIALOG(R)File 350:Derwent WPIX

015669688 **Image available**
WPI Acc No: 2003-731875/200369
XRPX Acc No: N03-584966

Data objects **persistence maintaining system, has set of definitions for relationship between data source schema and objects and programming module containing logic capable of persisting objects**

Patent Assignee: MULLINS W (MULL-I); THOUGHT INC (THOU-N)
Inventor: MULLINS W

Number of Countries: 103 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 200377113	A1	20030918	WO 2003US6987	A	20030307	200369	B
AU 2003220077	A1	20030922	AU 2003220077	A	20030307	200431	

Priority Applications (No Type Date): US 2002362345 P 20020307
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200377113	A1	E	86	G06F-007/00	

Designated States (National): AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
Designated States (Regional): AT BE BG CH CY CZ DE DK EA EE ES FI FR GB GH GM GR HU IE IT KE LS LU MC MW MZ NL OA PT SD SE SI SK SL SZ TR TZ UG ZM ZW

AU 2003220077 A1 G06F-007/00 Based on patent WO 200377113

Data objects **persistence maintaining system, has set of definitions for relationship between data source schema and objects...**

Abstract (Basic):

... schema and objects (1,10,20,30,40,50,60,70,80,90) capable of **storing data** for an **object** language application. A programming module contains logic capable of persisting an indicated **object** or set of **objects** . The persisted **data** are **stored** in a **data** source. An input method informs the programming module about the location of the objects.

... for creating or maintaining transparent persistence of a unit selected from the group consisting of **data objects** .

...

...Used for creating and maintaining persistence of **data objects** and associated **data stores** .

...

...The system creates and maintains transparent persistence of **complex data objects** without the need for inserting **byte** codes or modification of object graphs. The system enables copies of a data graph to...

...The drawing shows a complex **data object** graph drawing illustrating a customer object and its related objects

International Patent Class (Main): **G06F-007/00**
International Patent Class (Additional): **G06F-017/00**

RELATED DOCUMENTS BENEATH

12/3,K/5 (Item 4 from file: 350)
DIALOG(R)File 350:Derwent WPIX

014037837
WPI Acc No: 2001-522050/200157
XRPX Acc No: N01-386917
Computerized method of managing binary large objects **in** database **management system by using** object **handler** storing objects **in contiguous** storage **section**
Patent Assignee: UNISYS CORP (BURS)
Inventor: BRUSO K L; CONNER R W
Number of Countries: 021 Number of Patents: 003
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
WO 200148638	A2	20010705	WO 2000US34813	A	20001221	200157 B
US 6615219	B1	20030902	US 99474552	A	19991229	200366
EP 1342173	A2	20030910	EP 2000986667	A	20001221	200367
			WO 2000US34813	A	20001221	

RELATED DOCUMENTS PENDING

Priority Applications (No Type Date): US 99474552 A 19991229
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 200148638	A2	E	34	G06F-017/30	
	Designated States (National): JP				
	Designated States (Regional): AT BE CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE TR				
US 6615219	B1			G06F-017/30	
EP 1342173	A2	E		G06F-017/30	Based on patent WO 200148638
	Designated States (Regional): DE GB				

Computerized method of managing binary large objects **in** database **management system by using** object **handler** storing objects **in contiguous** storage **section**

Abstract (Basic):
... Method consists in constructing a database table with rows of **data** which include **fixed - length data** elements and **object** identifiers referencing the binary **objects** . Each object is **stored** in a contiguous storage section referenced by the associated identifier. The data rows are read...
... There is an INDEPENDENT CLAIM for an apparatus for managing binary **large objects** in a database management system...
International Patent Class (Main): **G06F-017/30**

12/3,K/6 (Item 5 from file: 350)
DIALOG(R)File 350:Derwent WPIX

013485622 **Image available**
WPI Acc No: 2000-657565/200064
XRPX Acc No: N00-487521
Modifying object **-oriented** database **transformation** process **by translating values from complex data to** primitive data types **for expressing one or more representing values in object-oriented database in terms of** primitive data types
Patent Assignee: SUN MICROSYSTEMS INC (SUNM)
Inventor: NELSON M R; SAULPAUGH T E; SLAUGHTER G L; TRAVERSAT B A
Number of Countries: 026 Number of Patents: 002
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 1030253	A1	20000823	EP 2000301176	A	20000215	200064	B
US 6609130	B1	20030819	US 99253867	A	19990219	200356	

Priority Applications (No Type Date): US 99253867 A 19990219
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 1030253	A1	E	35	G06F-017/30	
US 6609130	B1			G06F-017/30	

Designated States (Regional): AL AT BE CH CY DE DK ES FI FR GB GR IE IT LI LT LU LV MC MK NL PT RO SE SI

Modifying object **-oriented** database **transformation** process **by translating values from complex data to** primitive data types **for expressing one or more representing values in object-oriented database in terms of** primitive data types

Abstract (Basic):
... A plug-in module is invoked which understands the **complex** data **type** and the primitive **data types** . The values are translated from the complex **data type** to the **primitive data types** for expressing the one or more values representing values in the object-oriented database in terms of the **primitive data types** .
... form into an intelligent intermediate form (350). The latter can be turned back into an **object** -oriented **database** (340) through the **process** of **database** population (352). A transformation customizer (354) can be used to extend the grammar to allow for the use of complex **data types** in serialization (342) and compilation...

...Provides an intelligent mechanism and **process** for **storing** an **object** -oriented configuration **database**

International Patent Class (Main): **G06F-017/30**

12/3,K/9 (Item 8 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011735100 **Image available**
WPI Acc No: 1998-152010/199814
XRPX Acc No: N98-121105
Variable length **data transfer system using ATM cell - has variable length packet which is divided in** pay load **of ATM cell in consideration with stored data** length
Patent Assignee: NEC CORP (NIDE)
Number of Countries: 001 Number of Patents: 001
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
JP 10023038	A	19980123	JP 96195630	A	19960705	199814	B

Priority Applications (No Type Date): JP 96195630 A 19960705
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
JP 10023038	A		10	H04L-012/28	

Variable length **data transfer system using ATM cell**...

...**has variable** length **packet which is divided in** pay load **of ATM cell in consideration with stored data** length

...Abstract (Basic): The system forwards several **variable length** packets to ATM cell stream. The **variable length** packet is divided in the ATM cell **pay load** (1-2-1-6) in consideration with the stored data **length** . Thus, the divided **variable length** packets are **stored** and forwarded **sequentially** .
...

...ADVANTAGE - Improves quality of forwarding data. Enlarges data **length** of individual **variable length** packets

12/3,K/10 (Item 9 from file: 350)
DIALOG(R)File 350:Derwent WPIX

011227659 **Image available**
WPI Acc No: 1997-205562/199719
XRPX Acc No: N97-169635

Processor with compiler-allocated, variable length **intermediate storage - in which intermediate storage can be allocated prior to run time for variable-sized** data objects **, and accessed through table of alias entries**

Patent Assignee: INT BUSINESS MACHINES CORP (IBMC); IBM CORP (IBMC)
Inventor: ENGEBRETSEN D R; GREGOR S L; MOUDGILL M; WILLIS J C
Number of Countries: 005 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week
EP 767424	A2	19970409	EP 96306526	A	19960909	199719 B
JP 9171461	A	19970630	JP 96220997	A	19960822	199736
US 5860138	A	19990112	US 95537556	A	19951002	199910
EP 767424	B1	20020417	EP 96306526	A	19960909	200227
DE 6920620702	E	20020523	DE 96620702	A	19960909	200241
			EP 96306526	A	19960909	
JP 3533294	B2	20040531	JP 96220997	A	19960822	200436

Priority Applications (No Type Date): US 95537556 A 19951002
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 767424	A2	E	19	G06F-009/35	
Designated States (Regional): DE FR GB					
JP 9171461	A		25	G06F-009/34	
US 5860138	A			G06F-009/26	
EP 767424	B1	E		G06F-009/35	
Designated States (Regional): DE FR GB					
DE 6920620702	E			G06F-009/35	Based on patent EP 767424
JP 3533294	B2		25	G06F-009/38	Previous Publ. patent JP 9171461

Processor with compiler-allocated, variable length **intermediate storage...**

...in which intermediate storage can be allocated prior to run time for variable-sized data objects **, and accessed through table of alias entries**

...Abstract (Basic): unit (30) having high- speed memory storage locations allocated at compile time for variable-sized **data objects** . The **storage** locations are accessed through a table of alias entries (34) that consist of a base...
...space that is encoded into relevant machine opcodes. The names are used to reference the **data objects** . The **processor** (20) can optionally include a **data** cache (28) and can be used in either single processor or multi-tasking environments. Reference...

...USE/ADVANTAGE - Buffer storage between processor core and main memory. Provides improved storage flexibility than **fixed length** storage of hardware registers and cache registers. Compiler for use with processor can allocate intermediate...
International Patent Class (Main): **G06F-009/26** ...

... **G06F-009/34** ...

... **G06F-009/35** ...

12/3,K/12 (Item 11 from file: 350)
DIALOG(R)File 350:Derwent WPIX

010165762 **Image available**
WPI Acc No: 1995-067015/199509
XRPX Acc No: N95-053193

Data management using nested records and code points - has implicit relationship defined by length fields of records and provides for movement of "chunks" of records hierarchically related

Patent Assignee: PARK CITY GROUP INC (PARK-N)
Inventor: BENNION H R; BENNION R H; BENNION H
Number of Countries: 055 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
WO 9502218	A1	19950119	WO 94US7686	A	19940701	199509	B
AU 9473585	A	19950206	AU 9473585	A	19940701	199518	
ZA 9404927	A	19960327	ZA 944927	A	19940707	199619	
EP 707724	A1	19960424	EP 94922508	A	19940701	199621	
			WO 94US7686	A	19940701		
US 5634123	A	19970527	US 9388788	A	19930708	199727	
CA 2166809	C	20000530	CA 2166809	A	19940701	200040	
			WO 94US7686	A	19940701		

Priority Applications (No Type Date): US 9388788 A 19930708
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
WO 9502218	A1	E	25	G06F-012/04	
Designated States (National): AT AU BB BG BR BY CA CH CN CZ DE DK ES FI GB GE HU JP KE KG KP KR KZ LK LU LV MD MG MN MW NL NO NZ PL PT RO RU SD SE SI SK TJ TT UA UZ VN					
Designated States (Regional): AT BE CH DE DK ES FR GB GR IE IT LU MC NL OA PT SE					
AU 9473585	A			G06F-012/04	Based on patent WO 9502218
ZA 9404927	A		30	G06F-000/00	
EP 707724	A1	E	1	G06F-012/04	Based on patent WO 9502218
Designated States (Regional): DE GB IT					
US 5634123	A		15	G06F-017/30	
CA 2166809	C	E		G06F-012/04	Based on patent WO 9502218

...Abstract (Basic): The data management system (100) stores and communicates different types of data and allows **variable lengths** and hierarchical nesting of data records. A hierarchical structure is implicity defined by relationships of...

...into several sections. Application programs (106) are stored in RAM in a conventional manner. Data **storage** (107) stores data for use by the computer, including **data objects** (110) organised according to the code point structure. Some of the data is stored on...

International Patent Class (Main): **G06F-000/00** ...

... **G06F-012/04** ...

... **G06F-017/30**

12/3,K/16 **(Item 15 from file: 350)**
DIALOG(R)File 350:Derwent WPIX

007963836 **Image available**
WPI Acc No: 1989-228948/198932
XRPX Acc No: N89-174677
Memory array storage **registers series** sequential **reading appts. - includes** data stream **counter tracking output bits issuing increment address signal to address latch at fixed count**
Patent Assignee: NAT SEMICONDUCTOR CORP (NASC); NAT SEMICONDUCTOR INC (NASC)
Inventor: BODDU S; KOWSHIK V; LUCERO E M
Number of Countries: 007 Number of Patents: 006
Patent Family:

Patent No	Kind	Date	Applicat No	Kind	Date	Week	
EP 326885	A	19890809	EP 89101064	A	19890121	198932	B
US 4873671	A	19891010	US 88149399	A	19880128	198950	
EP 326885	A3	19920513	EP 89101064	A	19890121	199330	
EP 326885	B1	19940928	EP 89101064	A	19890121	199437	
CA 1332470	C	19941011	CA 589315	A	19890127	199441	
DE 68918469	E	19941103	DE 618469	A	19890121	199443	
			EP 89101064	A	19890121		

Priority Applications (No Type Date): US 88149399 A 19880128
Patent Details:

Patent No	Kind	Lan	Pg	Main IPC	Filing Notes
EP 326885	A	E	8		
Designated States (Regional): DE FR GB IT NL					
US 4873671	A		7		
EP 326885	B1	E	9	G11C-007/00	
Designated States (Regional): DE FR GB IT NL					
DE 68918469	E			G11C-007/00	Based on patent EP 326885
CA 1332470	C			G11C-007/00	

Memory array storage **registers series** sequential **reading appts...**

...**includes** data stream **counter tracking output bits issuing increment address signal to address latch at fixed count**

...Abstract (Basic): An address register/counter latches starting address input from instruction shift register. A **data stream** counter monitors the number of clock pulses during the READ mode and generates an increment address latch (IAL) signal at a fixed count. The **data stream** counter also generates a signal during which time a new register in memory array corresponding...
...Abstract (Equivalent): An address register/counter latches starting address input from instruction shift register. A **data stream** counter monitors the number of clock pulses during the READ mode and generates an increment address latch (IAL) signal at a fixed count. The **data stream** counter also generates a signal during which time a new register in memory array corresponding...

..EP-326885 Serial read access circuitry for reading **sequential storage** registers within a memory array (2, 24) that includes a plurality of storage registers organised for read access by having **sequential** binary addresses associated therewith, each **storage** register having capacity for storing data comprising a plurality of data bits, the serial read...

// equivalent US 4873671, included HEREIN

...and provides the received data bits as an output in serial form; and (9) a **data stream** counter (16) that counts the number of data bits provided by the data shift register...

...the data shift register, whereby the serial read access circuitry initiates a read of a **sequence** of **storage** registers in the array, the read sequence comprising a variable number of **storage** registers ...

...Abstract (Equivalent): sequential access functions and allows the memory to be used as a shift register of **variable length** .

(

```
Set     Items   Description
S1       5060   SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR -
            CLRSERIALIZ? OR CLRSERIALIS?
S2      12476   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
            ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3     757202   FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT?
             OR INCHOAT?
S4     471969   SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM
            OR BYTE?)
S5     113769   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
            OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S6       6570   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7      37380   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
            DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
            M)
S8       4711   DATAOBJECT? OR DATA()OBJECT?
S9        639   DATAMEMBER? OR DATA()MEMBER?
S10     22963   (RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11    154221   (LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PRED-
            ICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED
             OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12    960742   ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR F-
            LANK? OR BESIDE OR CLOSE OR CONTIGU?
S13    115330   INSTANTIAT? OR UPDAT? OR UP()(DATE? OR DATING?)
S14    143713   IC=G06F?
S15       367   S1:S2(10N)S3
S16        79   S1:S2(10N)S5
S17       433   S15:S16
S18       249   S17 AND S14
S19       337   S17 AND S7:S11
S20        83   S19 AND S7 AND S8:S9
S21         4   S17 AND S8:S9(5N)S12(5N)S4
S22         0   S19 AND (S1 AND S3)/TI
S23        37   S17 AND (S1 OR S3:S5)/TI
S24        13   S15 AND S16
S25        71   S18 AND S10
S26        67   S25 AND S4
S27        55   S26 AND S11
S28       140   S20:S21 OR S23:S24 OR S27
S29         4   AD=2005
S30       140   S28 NOT S29
S31       140   IDPAT (sorted in duplicate/non-duplicate order)
? show files
File 348:EUROPEAN PATENTS 1978-2005/May W01
         (c) 2005 European Patent Office
File 349:PCT FULLTEXT 1979-2005/UB=20050505,UT=20050428
         (c) 2005 WIPO/Univentio
?
```

Set	Items	Description
S1	5060	SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR CLRSERIALIZ? OR CLRSERIALIS?
S2	12476	(STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR CONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3	757202	FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT? OR INCHOAT?
S4	471969	SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM OR BYTE?)
S5	113769	(HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S6	6570	LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7	37380	(FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATUM)
S8	4711	DATAOBJECT? OR DATA()OBJECT?
S9	639	DATAMEMBER? OR DATA()MEMBER?
S10	22963	(RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11	154221	(LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PREDICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12	960742	ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR FLANK? OR BESIDE OR CLOSE OR CONTIGU?
S13	115330	INSTANTIAT? OR UPDAT? OR UP()(DATE? OR DATING?)
S14	143713	IC=G06F?
S15	367	S1:S2(10N)S3
S16	79	S1:S2(10N)S5
S17	433	S15:S16
S18	249	S17 AND S14
S19	337	S17 AND S7:S11
S20	83	S19 AND S7 AND S8:S9
S21	4	S17 AND S8:S9(5N)S12(5N)S4
S22	0	S19 AND (S1 AND S3)/TI
S23	37	S17 AND (S1 OR S3:S5)/TI
S24	13	S15 AND S16
S25	71	S18 AND S10
S26	67	S25 AND S4
S27	55	S26 AND S11
S28	140	S20:S21 OR S23:S24 OR S27
S29	4	AD=2005
S30	140	S28 NOT S29
S31	140	IDPAT (sorted in duplicate/non-duplicate order)
S32	34	S17 AND S1:S2(5N)FRAGMENT?
S33	24	S32 NOT S28
S34	24	IDPAT (sorted in duplicate/non-duplicate order)

SUPPLEMENTAL STRATEGY

31/3/5 (Item 5 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01068149

COMPUTER SYSTEM FOR TRANSFERRING MULTIPLE HIGH BANDWIDTH STREAMS OF DATA BETWEEN MULTIPLE STORAGE UNITS AND MULTIPLE APPLICATIONS IN A SCALABLE AND RELIABLE MANNER

COMPUTERSYSTEM FUR EINE SICHERE UND SKALIERBARE UBERTRAGUNG VON MEHRFACHDATENSTROME MIT HOHERER BANDBREITE ZWISCHEN MEHRFACHDATENEINHEITEN UND MEHRFACHAPPLIKATIONEN

SYSTEME INFORMATIQUE DE TRANSFERT DE MULTIPLES TRAINS DE DONNEES A GRANDE LARGEUR DE BANDE ENTRE DE MULTIPLES UNITES DE STOCKAGE ET DE MULTIPLES APPLICATIONS, DE MANIERE FIABLE ET MODULABLE

PATENT ASSIGNEE:
AVID TECHNOLOGY, INC., (1306171), Metropolitan Technology Park, One Park West, Tewksbury, MA 01876, (US), (Proprietor designated states: all)
INVENTOR:
PETERS, Eric, C., 80 Carleton Road, Carlisle, MA 01741, (US)
RABINOWITZ, Stanley, 12 Vine Brook Road, Westford, MA 01886, (US)
JACOBS, Herbert, R., 17 Sunrise Drive, Hudson, NH 03051, (US)
GILLETT, Richard, Baker, Jr., 30 Preservation Way, Westford, MA 01886, (US)
FASCIANO, Peter, J., 137Everett street, Natick, MA 01760, (US)
LEGAL REPRESENTATIVE:
Kazi, Ilya et al (86111), Mathys & Squire, 100 Gray's Inn Road, London WC1X 8AL, (GB)

PATENT (CC, No, Kind, Date): EP 1040419 A1 001004 (Basic)
EP 1040419 B1 020807
WO 9934291 990708
APPLICATION (CC, No, Date): EP 98964190 981221; WO 98US27199 981221
PRIORITY (CC, No, Date): US 997769 971224; US 6070 980112; US 54761 980403
DESIGNATED STATES: DE; FR; GB; NL
RELATED DIVISIONAL NUMBER(S) - PN (AN):
EP 1217557 (EP 2002002149)
INTERNATIONAL PATENT CLASS: G06F-011/20; G06F-011/10; H04N-007/173
NOTE:
No A-document published by EPO
LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS B	(English)	200232	1152
CLAIMS B	(German)	200232	1207
CLAIMS B	(French)	200232	1338
SPEC B	(English)	200232	17415
Total word count - document A			0
Total word count - document B			21112
Total word count - documents A + B			21112

RELATED Doc. BENSMITH

31/3/18 (Item 18 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01829728
Modular object serialization **architecture**
Modulare Objektserialisierungsarchitektur
Architecture modulaire de serialisation **des objets**
PATENT ASSIGNEE:
MICROSOFT CORPORATION, (749866), One Microsoft Way, Redmond, WA 98052, (US), (Applicant designated States: all)
INVENTOR:
Pepin, Brian Keith, 1203 5th Avenue N, Seattle WA 98109, (US)
Burke, Shawn Patrick, 8932 123rd Lane NE, Kirkland WA 98033, (US)
LEGAL REPRESENTATIVE:
Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721), Maximilianstrasse 58, 80538 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 1489495 A2 041222 (Basic)
APPLICATION (CC, No, Date): EP 2004012786 040528;
PRIORITY (CC, No, Date): US 600256 030619
DESIGNATED STATES: AT; BE; BG; CH; CY; CZ; DE; DK; EE; ES; FI; FR; GB; GR; HU; IE; IT; LI; LU; MC; NL; PL; PT; RO; SE; SI; SK; TR
EXTENDED DESIGNATED STATES: AL; HR; LT; LV; MK
INTERNATIONAL PATENT CLASS: G06F-009/44
ABSTRACT WORD COUNT: 148
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200452	2253
SPEC A	(English)	200452	4547
Total word count - document A			6800
Total word count - document B			0
Total word count - documents A + B			6800

RELATED DOCUMENTS BENEATH

31/3/28 (Item 28 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01278054
Version-adaptive serialization **and deserialization of program objects**
Versionsadaptive Serialisierung **und Deserialisierung von Programmobjekten**
Seriation et deseriation adaptive des objets de logiciel
PATENT ASSIGNEE:
SEIKO EPSON CORPORATION, (730002), 4-1, Nishi-shinjuku 2-chome, Shinjuku-ku, Tokyo 163, (JP), (Applicant designated States: all)
INVENTOR:
Heistermann, Horst, 1055 Manet Drive, No.7, Sunnyvale, California 94087, (US)
Chia-Hsin, Li, 4521 Elmhurst Drive, San Jose, California 95129, (US)
LEGAL REPRESENTATIVE:
Grunecker, Kinkeldey, Stockmair & Schwanhausser Anwaltssozietat (100721), Maximilianstrasse 58, 80538 Munchen, (DE)
PATENT (CC, No, Kind, Date): EP 1100005 A2 010516 (Basic)
EP 1100005 A3 040630
APPLICATION (CC, No, Date): EP 2000116794 000803;
PRIORITY (CC, No, Date): US 410363 990930
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-009/44; G06F-009/46
ABSTRACT WORD COUNT: 99
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200120	2696
SPEC A	(English)	200120	6582
Total word count - document A			9278
Total word count - document B			0
Total word count - documents A + B			9278

RELATED DOCUMENT BENEATH

31/3/29 (Item 29 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01181955
Transformation customizer for configuration database compilation and serialization processes
Anpassung der Umwandlung beim komilieren und Serialisieren von Konfigurationsdatenbanken
L'adaption de conversion pour la compilation et sesialisation d'une base de donnees de configuration
PATENT ASSIGNEE:
SUN MICROSYSTEMS, INC., (1392733), 901 San Antonio Road, Palo Alto, California 94303, (US), (Applicant designated States: all)
INVENTOR:
Saulpaugh, Thomas E., 6938 Bret Harte Drive, San Jose, California 95120, (US)
Slaughter, Gregory L., 3326 Emerson Street, Palo Alto, California 94306, (US)
Traversat, Bernard A., 2055 California Street, Apt 402, San Francisco, California 94109, (US)
Nelson, Matthew R., 956 Kintyre Way, Sunnyvale, California 94087, (US)
LEGAL REPRESENTATIVE:
Harris, Ian Richard (72231), D. Young & Co., 21 New Fetter Lane, London EC4A 1DA, (GB)
PATENT (CC, No, Kind, Date): EP 1030253 A1 000823 (Basic)
APPLICATION (CC, No, Date): EP 301176 000215;
PRIORITY (CC, No, Date): US 253867 990219
DESIGNATED STATES: DE; FR; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: G06F-017/30
ABSTRACT WORD COUNT: 211
NOTE:
Figure number on first page: 5

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:

Available Text	Language	Update	Word Count
CLAIMS A	(English)	200034	1585
SPEC A	(English)	200034	10858
Total word count - document A			12443
Total word count - document B			0
Total word count - documents A + B			12443

RELATED DOCUMENT BELOW

31/3/32 (Item 32 from file: 348)
DIALOG(R)File 348:EUROPEAN PATENTS

01103863
System and method for storing and retrieving objects
Vorrichtung und Verfahren zum Abspeichern und Wiederauffinden von Objekten
Dispositif et procede pour le stockage et la recuperation des objets
PATENT ASSIGNEE:
Intellution Inc., (1243631), One Edgewater Drive, Norwood, Massachusetts 02062, (US), (Applicant designated States: all)
INVENTOR:
Gendron, Robert F, 24 Summer Road, Salem, Massachusetts 01970, (US)
Jones, Stephen Kent, 15 Wamer Way, Canton, Massachusetts 02021, (US)
LEGAL REPRESENTATIVE:
Garratt, Peter Douglas et al (43121), Mathys & Squire 100 Grays Inn Road, London WC1X 8AL, (GB)
PATENT (CC, No, Kind, Date): EP 967546 A2 991229 (Basic)
EP 967546 A3 010502
APPLICATION (CC, No, Date): EP 99305025 990625;
PRIORITY (CC, No, Date): US 90655 980625
DESIGNATED STATES: DE; GB
EXTENDED DESIGNATED STATES: AL; LT; LV; MK; RO; SI
INTERNATIONAL PATENT CLASS: **G06F-009/44** ; **G06F-009/46**
ABSTRACT WORD COUNT: 101
NOTE:
Figure number on first page: 1

LANGUAGE (Publication,Procedural,Application): English; English; English
FULLTEXT AVAILABILITY:
Available Text Language Update Word Count
CLAIMS A (English) 199952 1156
SPEC A (English) 199952 10652
Total word count - document A 11808
Total word count - document B 0
Total word count - documents A + B 11808

RELATED DOCUMENTS BENEATH

31/3/68 (Item 68 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

01043393 **Image available**
ITERATIVE SERIALISATION **PROCEDURE FOR STRUCTURED SOFTWARE OBJECTS**
PROCEDURE DE SERIALISATION **ITERATIVE POUR OBJETS LOGICIELS STRUCTURES**
Patent Applicant/Assignee:
AXALTO SA, 50, avenue Jean Jaures, F-92120 Montrouge, FR, FR (Residence), FR (Nationality), (For all designated states except: US)
SCHLUMBERGER MALCO INC, 9800 Reistertown Road, Owings Mills, MD 21117, US , US (Residence), US (Nationality), (Designated only for: MC)
Patent Applicant/Inventor:
FAMBON Olivier, 2 rue du Commandant Gillot, F-38000 Grenoble, FR, FR (Residence), FR (Nationality), (Designated only for: US)
FREYSSINET Andre, Le Sorbier, F-38760 Saint Paul de Varces, FR, FR (Residence), FR (Nationality), (Designated only for: US)
LACOURTE Serge, Le Soleil Levant, 2 bld des Anciens d'Algerie, F-38580 Allevars les Bains, FR, FR (Residence), FR (Nationality), (Designated only for: US)
Legal Representative:
AXALTO SA (commercial rep.), c/o Patricia RENAULT, 36-38 Rue de la Princesse, BP 45, F-78431 Louveciennes, FR,
Patent and Priority Information (Country, Number, Date):
Patent: WO 200373390 A2-A3 20030904 (WO 0373390)
Application: WO 2003IB763 20030226 (PCT/WO IB03000763)
Priority Application: FR 20022570 20020228
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW
(EP) AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT SE SI SK TR
(OA) BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG
(AP) GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW
(EA) AM AZ BY KG KZ MD RU TJ TM
Publication Language: English
Filing Language: English
Fulltext Word Count: 19706

RELATED DOCUMENTS BENEATH

31/3/120 (Item 120 from file: 349)
DIALOG(R)File 349:PCT FULLTEXT

00523472 **Image available**

DYNAMIC ALLOCATION FOR EFFICIENT MANAGEMENT OF VARIABLE SIZED DATA WITHIN A NONVOLATILE MEMORY

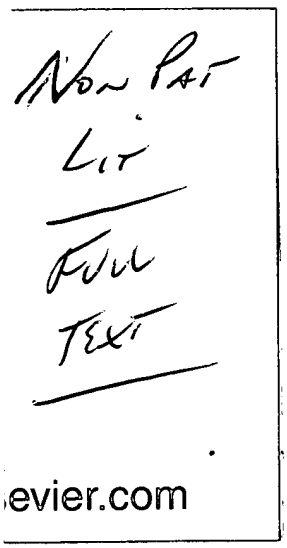
ATTRIBUTION DYNAMIQUE, AUX FINS DE GESTION EFFICACE, DE DONNEES DE DIMENSIONS VARIABLES DANS UNE MEMOIRE NON VOLATILE

Patent Applicant/Assignee:
INTEL CORPORATION,
SEE Deborah L,
HASBUN Robert N,
DUNLAP Jeffrey A,
DEL POZO Phillip J III,
Inventor(s):
SEE Deborah L,
HASBUN Robert N,
DUNLAP Jeffrey A,
DEL POZO Phillip J III,

Patent and Priority Information (Country, Number, Date):
Patent: WO 9954824 A1 19991028
Application: WO 99US8701 19990420 (PCT/WO US9908701)
Priority Application: US 9863954 19980421
Designated States:
(Protection type is "patent" unless otherwise stated - for applications prior to 2004)
AE AL AM AT AU AZ BA BB BG BR BY CA CH CN CU CZ DE DK EE ES FI GB GD GE
GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MD MG MK
MN MW MX NO NZ PL PT RO RU SD SE SG SI SK SL TJ TM TR TT UA UG US UZ VN
YU ZA ZW GH GM KE LS MW SD SL SZ UG ZW AM AZ BY KG KZ MD RU TJ TM AT BE
CH CY DE DK ES FI FR GB GR IE IT LU MC NL PT SE BF BJ CF CG CI CM GA GN
GW ML MR NE SN TD TG
Publication Language: English
Fulltext Word Count: 9278

RELATED DOC. BENEATH

```
Set     Items   Description
S1      27482   SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR -
             CLRSERIALIZ? OR CLRSERIALIS?
S2      15504   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
             ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3    4676616   FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT?
              OR INCHOAT? OR SNIPPET?
S4    1132279   SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM
             OR BYTE?)
S5     140139   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
             OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S6      35030   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7     104814   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
             DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
             M)
S8       9491   DATAOBJECT? OR DATA()OBJECT?
S9       1636   DATAMEMBER? OR DATA()MEMBER?
S10    102778   (RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11   1323061   (LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PRED-
             ICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED
              OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12  11499952   ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR F-
             LANK? OR BESIDE OR CLOSE OR CONTIGU?
S13       274   S1:S2(10N)(S3 OR S5)
S14        89   S13 AND (S4 OR S6:S9)
S15       133   S13 AND S1(10N)S3
S16        16   S13 AND S11
S17         8   S13 AND S12(10N)(S4 OR S6:S9)
S18       207   S14:S17
S19        95   S18 AND (S14 OR S16:S17)
S20        95   S19 AND PY<2005
S21        64   RD (unique items)
? show files
File   9:Business & Industry(R) Jul/1994-2005/May 11
         (c) 2005  The Gale Group
File  13:BAMP 2005/May W1
         (c) 2005  The Gale Group
File  15:ABI/Inform(R) 1971-2005/May 12
         (c) 2005 ProQuest Info&Learning
File  16:Gale Group PROMT(R) 1990-2005/May 11
         (c) 2005 The Gale Group
File  20:Dialog Global Reporter 1997-2005/May 12
         (c) 2005 The Dialog Corp.
File  47:Gale Group Magazine DB(TM) 1959-2005/May 12
         (c) 2005 The Gale group
File  75:TGG Management Contents(R) 86-2005/May W1
         (c) 2005 The Gale Group
File  88:Gale Group Business A.R.T.S. 1976-2005/May 11
         (c) 2005 The Gale Group
File  98:General Sci Abs/Full-Text 1984-2004/Dec
         (c) 2005 The HW Wilson Co.
File 141:Readers Guide 1983-2005/Dec
         (c) 2005 The HW Wilson Co
File 148:Gale Group Trade & Industry DB 1976-2005/May 12
         (c)2005 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 239:Mathsci 1940-2005/Jun
         (c) 2005 American Mathematical Society
File 275:Gale Group Computer DB(TM) 1983-2005/May 12
         (c) 2005 The Gale Group
```



File 369:New Scientist 1994-2005/Apr W1
(c) 2005 Reed Business Information Ltd.
File 370:Science 1996-1999/Jul W3
(c) 1999 AAAS
File 484:Periodical Abs Plustext 1986-2005/May W2
(c) 2005 ProQuest
File 553:Wilson Bus. Abs. FullText 1982-2004/Dec
(c) 2005 The HW Wilson Co
File 610:Business Wire 1999-2005/May 12
(c) 2005 Business Wire.
File 613:PR Newswire 1999-2005/May 12
(c) 2005 PR Newswire Association Inc
File 621:Gale Group New Prod.Annou.(R) 1985-2005/May 12
(c) 2005 The Gale Group
File 624:McGraw-Hill Publications 1985-2005/May 11
(c) 2005 McGraw-Hill Co. Inc
File 634:San Jose Mercury Jun 1985-2005/May 11
(c) 2005 San Jose Mercury News
File 635:Business Dateline(R) 1985-2005/May 12
(c) 2005 ProQuest Info&Learning
File 636:Gale Group Newsletter DB(TM) 1987-2005/May 12
(c) 2005 The Gale Group
File 647:CMP Computer Fulltext 1988-2005/Apr W4
(c) 2005 CMP Media, LLC
File 674:Computer News Fulltext 1989-2005/May W2
(c) 2005 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2005/May 11
(c) 2005 The Dialog Corp.
File 810:Business Wire 1986-1999/Feb 28
(c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
(c) 1999 PR Newswire Association Inc
?

21/3,K/24 (Item 1 from file: 88)
DIALOG(R)File 88:Gale Group Business A.R.T.S.

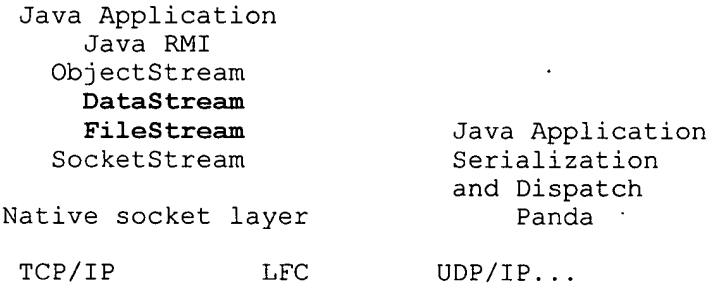
06182073 SUPPLIER NUMBER: 87509976
Efficient Java RMI for parallel programming.
Maassen, Jason; Van Nieuwpoort, Rob; Veldema, Ronald; Bal, Henri; Kielmann, Thilo; Jacobs, Ceriel; Hofman, Rutger
ACM Transactions on Programming Languages & Systems, 23, 6, 747(29)
Nov, 2001
ISSN: 0164-0925 LANGUAGE: English RECORD TYPE: Fulltext; Abstract
WORD COUNT: 13325 LINE COUNT: 01145

... heterogeneity, and flexibility (Waldo 1998).
Unfortunately, many existing Java implementations have inferior performance of both **sequential** code and communication primitives, which is a serious disadvantage for high-performance computing. Much effort is being invested in improving **sequential** code performance by replacing the original bytecode interpretation scheme with just-in-time compilers, native ...

...arguments (i.e., converting them to arrays of bytes) is implemented by recursively inspecting object **types** until **primitive types** are reached, and then invoking the **primitive serializers** . All of this is performed at runtime for each remote invocation.
Besides inefficiencies in the...

...much as possible by compile-time analysis. Manta uses a native compiler to generate efficient **sequential** code and specialized serialization routines for serializable argument classes. Also, Manta sends type descriptors for...system for the Manta RMI protocol is written in C. It was designed to minimize **serialization** and dispatch overhead such as copying, buffer management, **fragmentation** , thread switching, and indirect method calls. Figure 2 gives an overview of the layers in...

...Sun and Manta RMI protocols; shaded layers run compiled code.



...s serialization protocol performs optimizations for simple objects. An array whose elements are of a **primitive type** is **serialized** by doing a direct memory copy into the LFC buffer, so the array need not...JDK is that Manta uses a native compiler, whereas the JDK uses a JIT. The **sequential** speed of the code generated by the Manta compiler is much better than that of...

...to the IBM JDK/JIT. The overhead of the Java Native Interface and differences in **sequential** code speed obscures the comparison between the Manta and Sun RMI protocols. To allow a...It also reduces the garbage collection overhead for objects passed to RMI calls. Finally, its

sequential code speed is much better than that of the Sun JDK JIT, and is comparable...be attributed to the small size of the nodes and the dynamic nature of this **data type** , which makes especially (de)serialization expensive: the tree is written to and read from the...each system are computed relative to the parallel Manta program on a single CPU. The **sequential** execution times of Manta and Sun compiled are very similar, as the applications are compiled...

20011101

21/3,K/50 (Item 11 from file: 275)
DIALOG(R)File 275:Gale Group Computer DB(TM)

01890604 SUPPLIER NUMBER: 17957206 (USE FORMAT 7 OR 9 FOR FULL TEXT)
Programming Windows 95 with MFC, part VII: the document/view architecture. (Microsoft Foundation Classes) (Technology Tutorial)(Technical)
Prosise, Jeff
Microsoft Systems Journal, v11, n2, p19(17)
Feb, 1996
DOCUMENT TYPE: Technical ISSN: 0889-9932 LANGUAGE: English
RECORD TYPE: Fulltext; Abstract
WORD COUNT: 9247 LINE COUNT: 00981

... is used to initialize each new document that is created, while OnOpenDocument initializes the unserialized **data members** of the document object when a new document is loaded from disk. In an SDI...

...by the framework when a document is loaded from disk. Override to reinitialize the unserialized **data members** of the document object before a new document is loaded.
DeleteContents Called by the framework...pDocument **data member** and makes that pointer accessible through the view's GetDocument member function. Just as a...turn, invalidates the view's client area to force a repaint. Use OnInitialUpdate to initialize **data members** of the view class and perform other view-related initializations on a per-document basis...DYNCREATE macro adds three members to the class declaration: a static **data member** whose type is CRuntime-Class, a virtual function named GetRuntimeClass, and a static function named...

...in your document class and using the provided CArchive object to serialize the document's **data members** , you provide all the support the framework needs to implement the Open, Save, and Save...has never been so easy.
Suppose the data in your document consists of two int **data members** named m...

...if it's being loaded. The CArchive class overloads the << and >> operators so that primitive **data types** such as BYTEs, WORDS, DWORDs, LONGs, ints, floats, and doubles can be streamed in and out easily. MFC **data types** such as CStrings and CRects can be written and read the same way. MFC 4.0 is the first version to support the serialization of the int **data type** directly; in the past, ints had to be cast to WORDs, DWORDs, or other types...

...dependent.
Entire classes can be made **serializable** just as **primitive data types** are **serializable** : by deriving a class from CObject, throwing in a few macros, and adding a Serialize function to serialize the class's **data members** . MFC builds serialization support into many of its classes, including the collection classes designed to...sense, as opposed to handling them all in the frame window class.
During the routine **sequence** , command messages sent to an SDI frame window follow the path in Figure 4. The...byGrid. CByteArray is a private **data member** , so it cannot be manipulated outside of its own class. The state of a cell...doesn't override CDocument::OnOpenDocument because loading a document from disk initializes all the necessary **data members** .

Life's view class, CLifeView, is derived from CScrollView so the view

can be scrolled...cy **data** **members** holding the grid's dimensions. The m
...

19960200

United States Patent [19]

Kowshik et al.

[11] **Patent Number: 4,873,671**

[45] **Date of Patent: Oct. 10, 1989**

[54] **SEQUENTIAL READ ACCESS OF SERIAL MEMORIES WITH A USER DEFINED STARTING ADDRESS**

[75] Inventors: **Vikram Kowshik, San Jose; Sudhakar Boddu, Sunnyvale; Elroy M. Lucero, San Jose, all of Calif.**

[73] Assignee: **National Semiconductor Corporation, Santa Clara, Calif.**

[21] Appl. No.: **149,399**

[22] Filed: **Jan. 28, 1988**

[51] **Int. Cl.**[4] **G11C 7/00; G11C 8/00**

[52] **U.S. Cl.** **365/189.12**; 365/230.09; 365/78; 377/76

[58] **Field of Search** 365/230, 189, 239, 236, 365/240, 78, 221; 377/76, 69, 70

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,138,732	2/1979	Suzuki et al.	364/900
4,236,255	11/1980	Jansen et al.	364/900
4,422,160	12/1983	Watanabe	365/189 X
4,507,760	3/1985	Fraser	365/221
4,646,270	2/1987	Voss	3265/221
4,751,684	6/1988	Holt	365/189

FOREIGN PATENT DOCUMENTS

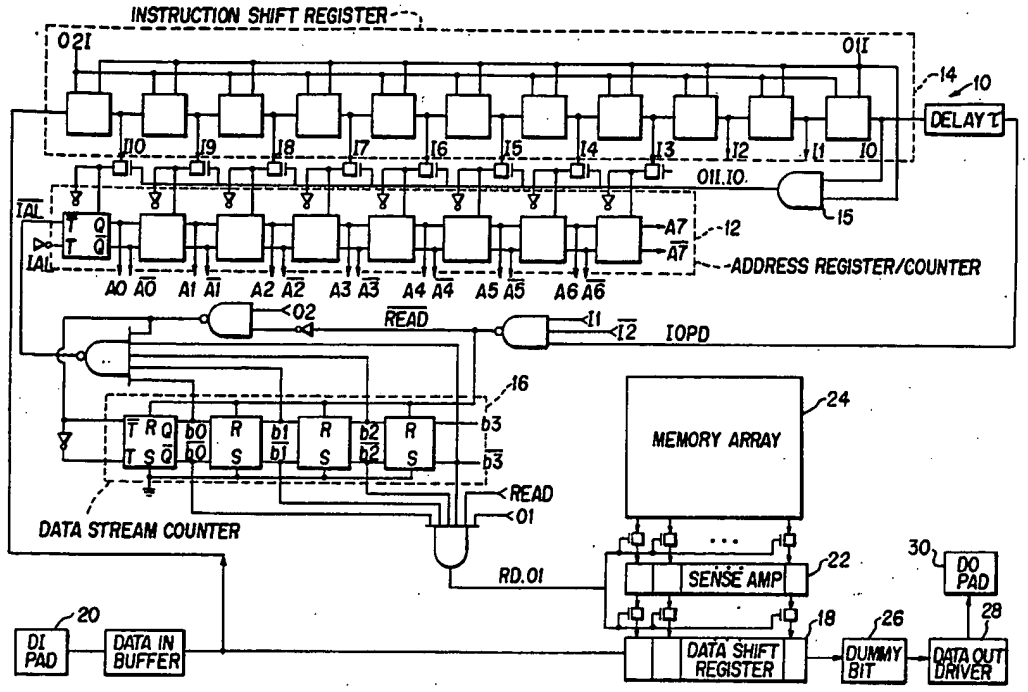
3634657	4/1987	Fed. Rep. of Germany	365/78
2183374	6/1987	United Kingdom .	

Primary Examiner—Stuart N. Hecker
Assistant Examiner—Alfonso Garcia
Attorney, Agent, or Firm—Limbach, Limbach & Sutton

[57] **ABSTRACT**

Circuitry for serial read memory access utilizing a random starting address is disclosed. Fast read access is provided without upsetting the original data pattern stored in the memory core if the sequential read is terminated in midstream. After the last memory address is reached, the access automatically rolls over to the first address. The circuit provides both random and sequential access functions and allows the memory to be used as a shift register of variable length.

4 Claims, 3 Drawing Sheets



On the seventeenth high going phase of clock signal $\phi 2$ following the decoding of the READ instruction, the MSB (data bit D15*) of the incremented address is shifted into the master of the dummy bit and at the seventeenth high going phase of $\phi 1$, the data bit D15* is output on the data output pad 30.

This sequence of events repeats for each 16 bits of data. Thus, a continuous stream of data bits can be serially clocked out of the data output pad 3 without the need for providing the new addresses to the circuit 10 by inputting new READ instructions over and over again. This provides a substantial time savings. In this way, the entire memory array 24 can be read in one continuous data stream or as registers of length varying from 16 to 4096 bits. Thus, the array 24 can also be used as a shift register of variable lengths (from 16 to 256 bits).

It should be understood that various alternatives to the embodiment of the invention shown herein may be employed in practicing the invention. It is intended that the following claims define the invention and that circuits within the scope of these claims and their equivalents be covered thereby.

What is claimed is:

1. Apparatus for reading a sequential series of storage registers within a memory array wherein the memory array includes a plurality of storage registers organized for read access by having sequential binary addresses associated therewith, each storage register having capacity for storing data comprising a plurality of data bits, the apparatus comprising:

(a) address register/counter means for storing a binary address which is used to access a preselected storage register within the memory array to serially read the data bits from the preselected storage register, the address register/counter means including means for incrementing the stored binary address by 1 upon receipt of an increment signal; and

(b) means for determining that all of the plurality of data bits stored in the preselected storage register have been read from the preselected storage register and for generating the increment signal in response to said determination such that data is read from storage registers within the memory array having sequential binary addresses, whereby the apparatus automatically initiates a read of a sequence of storage registers in the array, the read sequence comprising a variable number of storage registers.

2. A method for reading a plurality of sequential data storage registers within a memory array, the method comprising the steps of:

(a) accessing a preselected storage register within the array utilizing a binary address corresponding to the preselected storage register,

(b) reading data from the preselected storage register;

(c) sensing that data has been read from the preselected storage register;

(d) upon sensing that data has been read from the preselected storage register, automatically incrementing by 1 the binary address utilized to access the preselected storage register; and

(e) repeating steps (a)-(d) above utilizing the incremented binary addresses to read each of a plurality of sequential data storage registers within the memory array, thereby initiating the read of sequence of storage registers in the array, the read sequence comprising a variable number of storage registers.

3. A method as in claim 2 wherein the sequence of incremented binary addresses is returned to the first address in the sequence after the Nth data storage register has been read such that all N registers in the memory array are used.

4. Serial read access circuitry for reading sequential storage registers within a memory array of the type that includes a plurality of storage registers organized for read access by having sequential binary addresses associated therewith, each storage register having capacity for storing data comprising a plurality of data bits, the serial read access circuitry comprising:

(a) an instruction shift register that serially receives a read instruction comprising a plurality of data bits, the read instruction including the binary address of a preselected storage register within the memory array, the instruction shift register including means responsive to receipt by the instruction shift register means of all of the plurality of data bits of the read instruction for generating a latch signal;

(b) an address register/counter that stores the binary address of a storage register to be read, the address register/counter including means for incrementing by 1 the binary address stored therein in response to an increment signal, the address register/counter being responsive to the latch signal for receiving the binary address of the preselected storage register from the instruction shift register as the address stored therein;

(c) a data shift register that receives the plurality of data bits stored in the preselected storage register and provides the received data bits as an output in serial form; and

(d) a data stream counter that counts the number of data bits provided by the data shift register and generates the increment signal when all of the plurality of data bits have been transferred from the data shift register, whereby the binary address stored in the address register/counter is incremented by 1 such that the storage register in the memory array having the next sequential address is read, whereby the serial read access circuitry automatically initiates a read of a sequence of storage registers in the array, the read sequence comprising a variable number of storage registers.

* * * * *

US005363097A

United States Patent [19]

Jan

[11] Patent Number: 5,363,097

[45] Date of Patent: Nov. 8, 1994

[54] **DIRECT SEQUENTIAL-BIT VARIABLE LENGTH DECODER**

[75] Inventor: **Yung-Jung Jan**, Keelung, Taiwan, Prov. of China

[73] Assignee: **Industrial Technology Research Institute**, Hsinchu, Taiwan, Prov. of China

[21] Appl. No.: **944,657**

[22] Filed: **Sep. 14, 1992**

[51] **Int. Cl.**[5] **H03M 7/40**

[52] **U.S. Cl.** **341/67**; 341/63

[58] **Field of Search** 341/63, 67, 65, 64, 341/106

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,420,771	12/1983	Pirsch	341/67
4,813,056	3/1989	Fedele	341/107
4,827,336	5/1989	Acampora et al.	358/135
5,060,242	10/1991	Arbeiter	341/67
5,138,315	8/1992	LeQueau et al.	341/67
5,220,325	6/1993	Ackland et al.	341/67
5,223,348	8/1993	Pollman et al.	341/67
5,225,832	7/1993	Wang et al.	341/67

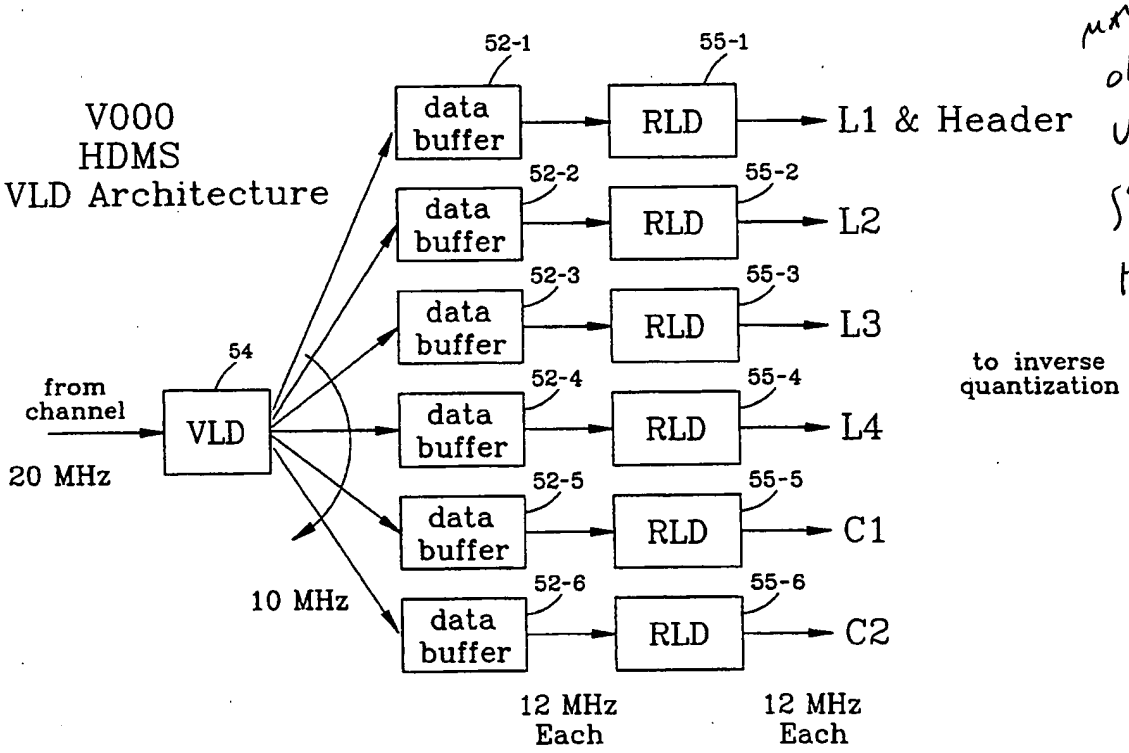
Primary Examiner—Howard L. Williams
Attorney, Agent, or Firm—Bo-In Lin

[57] **ABSTRACT**

The present invention comprises a high definition television (HDTV) receiver receiving a plurality of video data for display. The HDTV receiver comprises a VLD to first decode each of the video data into a fix-length data. The HDTV receiver further comprises a plurality of data memory banks for storing in parallel the fix-length data, and a plurality of run-length decoders (RLDs) to process in parallel the fix-length video data from the memory banks.

6 Claims, 5 Drawing Sheets

DIFFERENT TECHNOLOGY, BUT MAY BE OF USE — SEE HI-LITED CLAIMS



through-put due to the limitation of a variable-length decoder as encountered in the prior art is therefore resolved by the present invention.

The architecture as disclosed in the present invention is applicable not only to the digital HDTV systems, it can also be utilized in any decoding system which involves decoding the data encoded with various types of variable-length code (VLC) and run-length code (RLC). Many types of multi-media application involving the process of compressed video, audio, and numerical data can all be decoded by use of the decoding system according to the present invention.

Although the present invention has been described in terms of the presently preferred embodiment, it is to be understood that such disclosure is not to be interpreted as limiting. Various alternations and modifications will no doubt become apparent to those skilled in the art after reading the above disclosure. Accordingly, it is intended that the appended claims be interpreted as covering all alternations and modifications as fall within the true spirit and scope of the invention.

I claim:

1. A receiver arrangement of a digital communication system for receiving a plurality of variable length code words in the form of serial bit-stream comprising:

a variable length decoder (VLD) for directly receiving said serial bit-stream and for sequentially decoding each of said plurality of variable length code words into fixed-length data;

a data storage means including a plurality of data banks for sequentially receiving and temporarily storing said fixed-length data therein;

a digital communication processing means including a plurality of parallel processing means, each connecting to a corresponding data bank, wherein each of said data banks connected in parallel between said variable length decoder (VLD) and said digital communication processing means via said parallel processing means; and

each of said plurality of parallel processing means processing in parallel said fixed-length data for generating a plurality of digital communication data at a higher rate than the speed of said sequential decoding performed by said variable length decoder (VLD) to perform a real-time communication function.

2. A high definition television (HDTV) receiver arrangement for receiving and processing a plurality of video data in variable length code words for display comprising:

a variable-length decoder (VLD) for directly receiving said video data in serial bit-stream and for sequentially decoding each of said plurality of variable length code words into a fixed length data;

a data storage means for sequentially receiving and temporarily storing said fixed-length data therein and said data storage means further including a plurality of data banks;

a video-display data processing means for utilizing said fixed-length data from said data storage means for processing and generating a plurality of video display data;

said video-display data processing means further includes a plurality parallel run-length decoding processing means for performing run-length decoding on said fixed-length data, each connecting to a corresponding data bank, wherein each of said data banks connected in parallel between said variable length decoder (VLD) and said video-display data processing means via said parallel run-length decoding processing means; and

said video-display data processing means processing said fixed-length data for generating said plurality of video-display data at a rate which is higher than the speed of said sequential decoding performed by said variable length decoder (VLD) to perform a real-time video-display function.

3. The high definition television (HDTV) receiver arrangement of claim 2 wherein:

said variable length decoder receives said video data in serial bit-stream at a rate of approximately 20 MHz and generating said fixed-length data at a burst rate of approximately 10 Mhz; and

said data storage means including six data banks and said video-display data processing means including six parallel run-length decoding processing means wherein each of said plurality data banks and parallel run-length decoding processing means has a burst system speed of approximately 12 MHz and said video-display data processing means have a constant system speed of approximately 70 Mhz.

4. The high definition television (HDTV) receiver arrangement of claim 3 wherein:

said data banks of said data storage means include dynamic random access memory (DRAM) storage means.

5. The high definition television (HDTV) receiver arrangement of claim 2 wherein:

said variable-length decoder (VLD) sequentially decoding each of said plurality of variable length code words into a fixed length data by utilizing a Huffman decode table.

6. A high definition television (HDTV) receiver arrangement for receiving and processing a plurality of video data in variable length code words for display comprising:

a variable-length decoder (VLD) for directly receiving said video data in serial bit-stream and for sequentially decoding each of said plurality of variable length code words into a fixed length data by utilizing a Huffman decoding table wherein said variable length decoder receives said video data in serial bit-stream at a rate of approximately 20 MHz and generating said fixed-length data at a burst rate of approximately 10 Mhz;

a data storage means for sequentially receiving and temporarily storing said fixed-length data therein and said data storage means further including six DRAM data banks;

a video-display data processing means for utilizing said fixed-length data from said data storage means for processing and generating a plurality of video display data;

said video-display data processing means further includes six parallel run-length decoding processing means for performing run-length decoding on said fixed-length data, each connecting to a corresponding data bank wherein each of said data banks connected in parallel between said variable length decoder (VLD) and said video-display data processing means via said parallel run-length decoding processing means;

each of said plurality data banks and parallel run-length decoding processing means has a burst system speed of approximately 12 MHz; and

said video-display data processing means processing said fixed-length data for generating said plurality of video-display data at a rate of approximately 70 MHz to perform a real-time video-display function.

* * * * *

```
Set     Items   Description
S1       5461   SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR -
            CLRSERIALIZ? OR CLRSERIALIS?
S2       9137   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
            ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3    2560955   FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT?
             OR INCHOAT? OR SNIPPET?
S4    1931668   SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM
            OR BYTE?)
S5      39898   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
            OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S6       3304   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7      36031   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
            DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
            M)
S8       4365   DATAOBJECT? OR DATA()OBJECT?
S9        158   DATAMEMBER? OR DATA()MEMBER?
S10     11701   (RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11    318890   (LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PRED-
            ICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED
             OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12   2757830   ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR F-
            LANK? OR BESIDE OR CLOSE OR CONTIGU?
S13       957   S1:S2 AND (S3 OR S5)
S14       277   S13 AND S1 AND S3
S15        38   S14 AND (S4 OR S6)
S16        22   S14 AND S7:S9
S17         3   S14 AND S10:S11
S18         0   S14 AND S12(10N)(S4 OR S7:S11)
S19         7   S14 AND S12
S20       136   S13 AND (S4 OR S6)
S21        84   S13 AND S7:S9
S22        19   S13 AND S10:S11
S23         6   S13 AND S12(10N)(S4 OR S7:S11)
S24        11   S20 AND S21
S25        99   S15:S19 OR S22:S24
S26        99   S25 AND PY<2005
S27        70   RD (unique items)
? show files
File   2:INSPEC 1969-2005/Apr W4
         (c) 2005 Institution of Electrical Engineers
File   6:NTIS 1964-2005/May W1
         (c) 2005 NTIS, Intl Cpyrght All Rights Res
File   8:Ei Compendex(R) 1970-2005/May W1
         (c) 2005 Elsevier Eng.  Info. Inc.
File  34:SciSearch(R) Cited Ref Sci 1990-2005/May W2
         (c) 2005 Inst for Sci Info
File  35:Dissertation Abs Online 1861-2005/Apr
         (c) 2005 ProQuest Info&Learning
File  62:SPIN(R) 1975-2005/Feb W4
         (c) 2005 American Institute of Physics
File  65:Inside Conferences 1993-2005/May W2
         (c) 2005 BLDSC all rts. reserv.
File  94:JICST-EPlus 1985-2005/Mar W3
         (c)2005 Japan Science and Tech Corp(JST)
File  95:TEME-Technology & Management 1989-2005/Apr W1
         (c) 2005 FIZ TECHNIK
File  99:Wilson Appl. Sci & Tech Abs 1983-2005/Apr
         (c) 2005 The HW Wilson Co.
File 111:TGG Natl.Newspaper Index(SM) 1979-2005/May 11
         (c) 2005 The Gale Group
```

File 144:Pascal 1973-2005/May W1
         (c) 2005 INIST/CNRS
File 256:TecInfoSource 82-2005/Mar
         (c) 2005 Info.Sources Inc
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
?

```
Set     Items   Description
S1       5461   SERIALIZ? OR SERIALIS? OR XMLSERIALIZ? OR XMLSERIALIS? OR -
            CLRSERIALIZ? OR CLRSERIALIS?
S2       9137   (STORE? OR STORING OR STORAG? OR PROCESS? OR CONVERT? OR C-
            ONVERS?)(3N)OBJECT?(3N)(BYTE? OR DATA?)
S3    2560955   FRAGMENT? OR SEGMENT? OR INCOMPLET? OR PARTIAL? OR TRUNCAT?
             OR INCHOAT? OR SNIPPET?
S4    1931668   SEQUENT? OR SEQUENC? OR SEQUEN?(3N)STOR?(3N)(DATA OR DATUM
            OR BYTE?)
S5      39898   (HEADER? OR TYPE? OR LENGTH?)(20N)(PAYLOAD? OR PAY()LOAD? -
            OR MEMBER? OR OBJECTMEMBER? OR PRIMITIV? OR CELL?()REFEREN?)
S6       3304   LOB OR LOBS OR LARGE()OBJECT? OR LARGEOBJECT?
S7      36031   (FILE? OR DATA?)()STREAM? OR FILESTREAM? OR DATASTREAM? OR
            DATATYPE? OR DATA()TYPE? OR COLLECT?()ELEMENT?()(DATA OR DATU-
            M)
S8       4365   DATAOBJECT? OR DATA()OBJECT?
S9        158   DATAMEMBER? OR DATA()MEMBER?
S10     11701   (RECORD? OR STORE? OR STORAG? OR STORING?)(3N)FORMAT?
S11    318890   (LOCAT? OR SITE? OR ADDRESS? OR PATH? OR MEMBER?)(5N)(PRED-
            ICT? OR IDENTIF? OR LABEL? OR TAG OR TAGS OR TAGGING OR TAGGED
             OR FLAG? OR BOOKMARK? OR EARMARK? OR TOKEN? OR ASSOCIAT?)
S12   2757830   ADJACENT? OR NEXT()"TO" OR ABUT? OR PROXIM? OR "NEAR" OR F-
            LANK? OR BESIDE OR CLOSE OR CONTIGU?
S13       957   S1:S2 AND (S3 OR S5)
S14       277   S13 AND S1 AND S3
S15        38   S14 AND (S4 OR S6)
S16        22   S14 AND S7:S9
S17         3   S14 AND S10:S11
S18         0   S14 AND S12(10N)(S4 OR S7:S11)
S19         7   S14 AND S12
S20       136   S13 AND (S4 OR S6)
S21        84   S13 AND S7:S9
S22        19   S13 AND S10:S11
S23         6   S13 AND S12(10N)(S4 OR S7:S11)
S24        11   S20 AND S21
S25        99   S15:S19 OR S22:S24
S26        99   S25 AND PY<2005
S27        70   RD (unique items)
S28        75   S14 AND (S1 OR S3)/TI
S29        60   S28 NOT S26
S30        42   RD (unique items)
S31        11   S13 AND (S1 AND S3)/TI
S32         0   S31 NOT (S29 OR S25)
?
```

SUPPLEMENTAL

27/3,K/52 (Item 8 from file: 35)
DIALOG(R)File 35:Dissertation Abs Online

1067183 ORDER NO: NOT AVAILABLE FROM UNIVERSITY MICROFILMS INT'L.
A STUDY OF THE AVAILABILITY AND SERIALIZABILITY **IN A DISTRIBUTED DATABASE SYSTEM**
Author: CHEUNG, DAVID WAI-LOK
Degree: PH.D.
Year: **1988**
Corporate Source/Institution: SIMON FRASER UNIVERSITY (CANADA) (0791)
Source: VOLUME 50/04-B OF DISSERTATION ABSTRACTS INTERNATIONAL.
PAGE 1499.

A STUDY OF THE AVAILABILITY AND SERIALIZABILITY **IN A DISTRIBUTED DATABASE SYSTEM**
Year: **1988**

Replication of **data objects** enhances the reliability and availability of a distributed database system. However, due to the inherent conflict between **serializability** and availability, if **serializability** is to be guaranteed in a partitioned database system, degradation of availability is inevitable. We first characterize **serializable** transaction executions in a partitioned database system, by means of a graph theoretical method. We...
...transaction distributions that satisfy the "weak uniformity assumption".
Since it is impossible to simultaneously achieve **serializability** and high availability in a general database system, we investigate database systems in which constraints are imposed on the read/write activity of the transactions. In particular, we propose a **fragmented** database system, in which transactions are classified as either local or global. This model can ...

...transaction is made to wait until it is known that the transaction has read consistent **data object** values from other sites. In both approaches, no global transaction blocks a local transaction. Moreover...

Academic Search Premier; Business Source Corporate; Computer Science Index; Computer Source; Internet and Personal Computing Abstracts; Information Science & Technology

Abstracts for *((SERIALIZ* OR SERIALIS*) AND FRAGMENT*)* Add this search to folder |

Searched: Display link to this search

1 - 2 of 2 Pages: 1 **Sort by :** Date Add (1-2)

The number of available results reflects the removal of duplicates.

1. Inside MFC serialization -- Typesafe serialization that's fast and flexible. By: Beveridge, Jim. Dr. Dobb's Journal: Software Tools for the Professional Programmer, October 1, 1995, Vol. 20 Issue 10, p62, 6; (*AN IPCA0444543*) Add

2. Achieving High Availability in Distributed Databases. By: Garcia-Molina, Hector; Kogan, Boris. IEEE Transactions on Software Engineering, Jul88, Vol. 14 Issue 7, p886, 11p, 2 charts, 8 diagrams; (*AN 14309365*) Add

1 - 2 of 2 Pages: 1 Add (1-2)

Top of Page

Title: **Inside MFC serialization -- Typesafe serialization that's fast and flexible.**

Authors: Beveridge, Jim

Source: Dr. Dobb's Journal: Software Tools for the Professional Programmer; October 1, 1995, Vol. 20 Issue 10, p62, 6

Document Type: Article

Subject Terms: C (Computer program language)
OBJECT-oriented methods (Computer science)
MEMORY
APPLICATION software -- Development
DOCUMENTATION

Geographic Terms: UNITED States

Author-Supplied Keywords: Microsoft Foundation Classes
Microsoft

Abstract: Focuses on the serialization mechanism in the Microsoft Foundation Classes (MFC), which is typesafe and strongly grounded in modern, object-oriented design theory. Reports that after creating multiple document types in the same application, whenever loading a file MFC would correctly create the right kind of document object and call the proper member function. Explains that in order to handle run-time type information, MFC creates a registry of classes in the application, and the types in this registry are not hardcoded in any table. Indicates that in creating an object in MFC, the context is based on information read from a serialized archive, and the construction of an object is cleanly separated from the memory allocation. Also considers creating types from a file, and optimizing archives. Concludes that MFC implements a fast, flexible, powerful, and typesafe serialization mechanism. Includes three code listings and four code fragments.

ISSN: 1044-789X

Accession Number: IPCA0444543

Persistent link to this record: http://search.epnet.com/login.aspx?direct=true&db=iqh&an=IPCA0444543

Database: Internet and Personal Computing Abstracts

Title: Achieving High Availability in Distributed Databases.

Authors: Garcia-Molina, Hector[1]
Kogan, Boris[1]

Source: IEEE Transactions on Software Engineering; Jul88, Vol. 14 Issue 7, p886, 11p, 2 charts, 8 diagrams

Document Type: Article

Subject Terms: *DATABASE design
*DATABASES
*DISTRIBUTED databases
*ELECTRONIC data processing -- Distributed processing
*SOFTWARE engineering
*SYSTEM design

Author-Supplied Keywords: database systems
distributed computing systems
fault tolerance
network partitions
Data availability

Abstract: A new approach is presented for managing distributed database systems in the face of communication failures and network partitions. The approach is based on the idea of dividing the database into fragments and assigning each fragment a controlling entity called agent. The goals achieved by this approach include high data availability and the ability to operate without promptly and correctly detecting partitions. Finally, a new correctness criterion for transaction execution, called fragmentwise serializability, is introduced. It is less strict than the conventional serializability, but is meaningful enough to be a valuable alternative for some applications. [*ABSTRACT FROM AUTHOR*]

Author Affiliations: [1]Department of Computer Science, Princeton University, Princeton, NJ

ISSN: 0098-5589

Accession Number: 14309365

Persistent link to this record: http://search.epnet.com/login.aspx?direct=true&db=aph&an=14309365

Database: Academic Search Premier

XML for Data: Reuse it or lose it, Part 3

Realize the benefits of reuse

Level: Intermediate

Kevin Williams (kevin@blueoxide.com)
CEO, Blue Oxide Technologies, LLC
08 Jul 2003

In the final installment of this three-part column, Kevin Williams looks at some of the ways you can take advantage of the reusable XML components that he defined in the previous two installments of this column. Designing XML with reusable components can, in many ways, create direct and indirect benefits; Kevin takes a quick look at some of the most important.

Contents:
Reusable XSLT components
Class-to-XML mapping (fragment serialization, deserialization)
Bringing XML and Web services together
Conclusion
Resources
About the author
Rate this article

Related content:
Reuse it or lose it, Part 1
Reuse it or lose it, Part 2

Subscriptions:
dW newsletters
dW Subscription (CDs and downloads)

This column builds on the philosophy of XML reuse I described in the first two columns, so if you haven't read those yet you might want to before diving into this one (see Resources).

The first benefit of using reusable components isn't necessarily a direct benefit of the design of XML structures that use components, but it is a natural outcome of the approach. To create components that can be reused, you need to capture solid semantics about those components. These semantics can be extended into the processing code itself to make the programmer's job easier. Take a look at the brief example in Listing 1. Suppose you have the following customer XML document:

Listing 1. Example customer XML document

```
<customer>
  <name>Amalgamated Widgets, Inc.</name>
  <contact>Fred Smith</contact>
  <phone>304-555-1212</phone>
</customer>
```

If you were to design this document as a single XML schema, you might not capture the semantics for each datapoint in the schema -- which would make it more difficult to write code to accurately process instances. (For example, is `contact` a name or an e-mail address? Is the order of the name *first name, middle name, last name* or *last name, comma, first name, middle name*?) On the other hand, if you choose to design this document using reusable XML components (datapoints for `name`, `contact`, and so on), you have already forced yourself to capture good semantics, because you can't reuse the components without knowing precisely the semantics of those components. This means that the processing software can take these semantic constraints as read and simplify the programmer's job.

Reusable XSLT components

Another natural benefit of the component-based approach to XML design is the ability to reuse XSLT fragments to ensure a standardized presentation of information across many different documents. Again, this is a natural outcome of capturing good semantics and reusing elements and attributes whenever possible. Suppose you have the following two documents:

Listing 2. Example customer and supplier XML documents with no reuse

```
<customer>
  <name>Amalgamated Widgets, Inc.></name>
  <contact>Fred Smith</contact>
  <phone>304-555-1212</phone>
</customer>

<supplier>
  <companyName>Sprockets Unlimited</companyName>
  <salesContact>Joe Jones</contact>
  <contactPhone>540-555-6789</phone>
</supplier>
```

In these examples no effort has been made to reuse content, so elements with identical semantic constraints (such as `name` and `companyName`) have different tag names. When it comes time to write stylesheets to present these documents (as HTML, for example), you now have to write one stylesheet that processes the `customer` document and another that processes the `supplier` document -- leading to duplication of effort, which is just what you're trying to avoid. Now suppose that these two documents were designed with reuse in mind:

Listing 3. Customer and supplier XML documents designed for reuse

```
<my:customer xmlns:my="http://mycompany.com/schemas">
  <my:companyName>Amalgamated Widgets, Inc.</my:companyName>
  <my:salesContactName>Fred Smith</my:salesContactName>
  <my:salesContactPhone>304-555-1212</my:salesContactPhone>
</my:customer>

<my:supplier xmlns:my="http://mycompany.com/schemas">
  <my:companyName>Sprockets Unlimited</my:companyName>
  <my:salesContactName>Joe Jones</my:salesContactName>
  <my:salesContactPhone>540-555-6789</my:salesContactPhone>
</my:supplier>
```

Note that this solution uses a namespace for all elements. Whether you think namespaces are a good idea or not, the proper use of them ensures that every element and attribute in your schema is uniquely and unambiguously defined -- the combination of a namespace URL and an element or attribute name can always be mapped back to a single semantic assertion about the meaning of that element or attribute. Listing 3 also reuses the datapoints in these structures. If you then want `companyName` to be represented as an `<H2>` field in your HTML wherever it appears, you can write one fragment of XSLT that styles it this way:

Listing 4. Stylesheet fragment to render a companyName element

```
<xsl:template xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
              xmlns:my="http://mycompany.com/schemas">
              match="my:companyName">
  <H2><xsl:value-of select="." /></H2>
</xsl:template>
```

This fragment can then be included in the stylesheets for `my:customer` and `my:supplier`:

Listing 5. Including the companyName stylesheet fragment in other stylesheets

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:include href="companyName.xslt" />
  ...
</xsl:stylesheet>
```

Now, when the word comes down that `companyName` needs to be `H3` instead of `H2`, you just have to change companyName.xslt and all of the other stylesheets automatically reflect the change. If you hadn't made a conscious effort to ensure that the company name had a consistent identity across the various documents where it appears, this change would be much more difficult.

Class-to-XML mapping (fragment serialization, deserialization)

The third benefit I want to discuss begins to appear when higher-order elements are reused. Suppose you have the following structure that represents a person across all your XML documents:

Listing 6. Sample person XML fragment

```
<my:person xmlns:my="http://mycompany.com/schemas">
  <my:personName>Kevin Williams</my:personName>
  <my:address1>123 Anywhere Street</my:address1>
  <my:city>Anytown</my:city>
  <my:state>WV</my:state>
  <my:postalCode>25532</my:postalCode>
  <my:country>USA</my:country>
</my:person>
```

If this structure is reused in many different places in your XML schemas, you can simplify your processing code by writing a class that maps directly onto this element structure. The class might have a `Parse` method that reads an XML fragment representing a person and decomposes it into public members of the class; it might also have a `Serialize` method that creates an appropriate XML fragment based on the public members. By creating these sorts of XML-aware classes, you can make it possible to reuse parsing and serialization code -- as long as you have properly reused the structures in your XML schemas. Here's an example (written in pseudocode; this approach is equally applicable to any object-oriented programming language):

Listing 7. Pseudocode class representing a Person object

```
Class: Person
  Public String personName
  Public String address1
  Public String address2
  Public String city
  Public String state
  Public String postalCode
  Public String country

  Public Method boolean Parse (String personElement)

  Public Method string Serialize ()
```

Depending on the type of information being manipulated, you might also want to include other methods that provide additional functionality in this class as well (such as a `Persist()` method, which stores the person in a database). It should be clear how designing for reuse at the beginning of the development effort leads to these sorts of benefits later.

Bringing XML and Web services together

An unfortunate trend among today's XML Web services developers is the tendency to think of Web services as a distinct platform from XML. This is due, in part at least, to the Web services wizards included in most of the recent integrated development environments. When a wizard can automatically generate WSDL and SOAP messages for whatever methods you have lying around in your objects, why bother thinking about the underlying XML? Well, as you might guess, the answer is simple: reuse. For example, suppose I have a method on a data object that allows me to read and write a person to my relational database:

Listing 8. Person class, revisited

```
Class: Person
  Public String personName
  Public String address1
  Public String address2
  Public String city
  Public String state
  Public String postalCode
  Public String country

  Public Method boolean Persist()
```

```
Public Method boolean RetrievePerson (int personID)
```

I could just expose the properties and methods on this object as Web services, but it wouldn't really be useful -- I'd have to make eight calls just to store a person (seven calls to the `set` methods on the various properties and one call to the `Persist` method. The key is to keep in mind that Web services should be thought of as outward facing, even if they are only being used to connect systems internally. Think of it as writing an API to your system: You wouldn't define an API that required calls to be made this way, would you? While you're at it, you should define your Web services to take advantage of the design work you have already done on your XML instances. The benefit should be clear -- one shared set of semantics means that consumers of your Web services have a clearly-defined, unambiguous definition of each part of the inbound and outbound Web service payload, and can use it more easily than if the parameters had cryptic names like `fn`. A better definition to expose as a Web service might look like this:

Listing 9. Person class with a better Web service interface

```
Class: Person

  Public Method boolean StorePerson(string personName,
      string address1, string address2, string city, string state,
      string postalCode, string country)
```

Conclusion

You'll find many benefits to designing XML schemas using reusable components. These benefits lead directly to shorter development cycles and simpler maintenance of code. If you are designing a large system with many different types of XML documents, taking the time to identify reusable components of those documents early in the development effort benefits that effort in the long term.

Resources

- Read the two previous XML for Data column installments on reuse: Part 1 provides an overview of XML reuse in enterprise-level solutions (*developerWorks*, March 2003), while Part 2 focuses on the types of components that can be reused in XML designs and provides examples of each in XML and XML Schema (April 2003).
- For more practical XSLT techniques, check out *The XSLT Cookbook* by Sal Mangano (O'Reilly and Associates).
- The xml.com Web site provides a good variety of articles with new ideas for XML developers.
- Find more XML resources on the *developerWorks* XML zone.
- Get IBM WebSphere Studio, a suite of tools that automate XML development, both in Java and in other languages. It is closely integrated with the WebSphere Application Server, but can also be used with other J2EE servers.
- Find out how you can become an IBM Certified Developer in XML and related technologies.

About the author

Kevin Williams is the CTO of Blue Oxide Technologies, LLC, a company that designs XML and Web service creation software. Visit their Web site at http://www.blueoxide.com. Kevin can be reached for comment at kevin@blueoxide.com.

Comments?

Submit feedback

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	7	terek-s$.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:48
L2	11	kalhan-a$.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:49
L3	11	ponnekanti-n$.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:49
L4	60	rangarajan-s$.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:49
L5	16	zwilling-m$.in.	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:50
L6	101	1 2 3 4 5	US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT	OR	ON	2005/05/12 08:50

US 20040267835 A1 US-PGPUB
US 20040267828 A1 US-PGPUB
US 20040267809 A1 US-PGPUB
US 20040199530 A1 US-PGPUB
US 20030029378 A1 US-PGPUB
US 20020099918 A1 US-PGPUB
US 20020078015 A1 US-PGPUB
US 20020018857 A1 US-PGPUB
US 6868418 B1 USPAT
US 6804700 B1 USPAT
US 6792432 B1 USPAT
US 6778977 B1 USPAT
US 6643753 B2 USPAT
US 6606626 B1 USPAT
US 6591269 B1 USPAT
US 6587881 B1 USPAT
US 6531184 B2 USPAT
US 6493701 B2 USPAT
US 6363387 B1 USPAT
US 6356887 B1 USPAT
US 6249792 B1 USPAT
JP 2002028548 A JPO
EP 1158365 A1 EPO
US 6868418 B DERWENT
US 20040267835 A DERWENT
US 20040267828 A DERWENT
US 20040267809 A DERWENT
US 20040225895 A DERWENT
US 20040199530 A DERWENT
US 6804700 B DERWENT
US 6792432 B DERWENT
US 6778977 B DERWENT
US 6606626 B DERWENT
US 6591269 B DERWENT
US 6587881 B DERWENT
US 20020099918 A DERWENT
US 20020078015 A DERWENT
US 6363387 B DERWENT
US 6249792 B DERWENT

```
Set     Items   Description
S1        872   AU=(TEREK S? OR TEREK, S? OR KALHAN A? OR KALHAN, A? OR PO-
              NNEKANTI N? OR PONNEKANTI, N? OR RANGARAJAN S? OR RANGARAJAN,
              S? OR ZWILLING M? OR ZWILLING, M?)
S2          0   SONER(2N)TEREK OR AJAY(2N)KALHAN OR NAGAVAMSI(2N)PONNEKANTI
              OR SRIKUMAR(2N)RANGARAJAN OR (MIKE OR MICHAEL)(2N)ZWILLING
S3     582308   SERIALIZ? OR SERIALIS? OR FRAGMENT?
S4          5   S1:S2 AND S3
S5          4   RD (unique items)
? show files
File   2:INSPEC 1969-2005/Apr W4
         (c) 2005 Institution of Electrical Engineers
File   6:NTIS 1964-2005/May W1
         (c) 2005 NTIS, Intl Cpyrght All Rights Res
File   8:Ei Compendex(R) 1970-2005/May W1
         (c) 2005 Elsevier Eng.  Info. Inc.
File  34:SciSearch(R) Cited Ref Sci 1990-2005/May W2
         (c) 2005 Inst for Sci Info
File  35:Dissertation Abs Online 1861-2005/Apr
         (c) 2005 ProQuest Info&Learning
File  62:SPIN(R) 1975-2005/Feb W4
         (c) 2005 American Institute of Physics
File  65:Inside Conferences 1993-2005/May W2
         (c) 2005 BLDSC all rts. reserv.
File  94:JICST-EPlus 1985-2005/Mar W3
         (c)2005 Japan Science and Tech Corp(JST)
File  95:TEME-Technology & Management 1989-2005/Apr W1
         (c) 2005 FIZ TECHNIK
File  99:Wilson Appl. Sci & Tech Abs 1983-2005/Apr
         (c) 2005 The HW Wilson Co.
File 111:TGG Natl.Newspaper Index(SM) 1979-2005/May 11
         (c) 2005 The Gale Group
File 144:Pascal 1973-2005/May W1
         (c) 2005 INIST/CNRS
File 256:TecInfoSource 82-2005/Mar
         (c) 2005 Info.Sources Inc
File 434:SciSearch(R) Cited Ref Sci 1974-1989/Dec
         (c) 1998 Inst for Sci Info
?
```

```
Set     Items   Description
S1         72   AU=(TEREK S? OR TEREK, S? OR KALHAN A? OR KALHAN, A? OR PO-
              NNEKANTI N? OR PONNEKANTI, N? OR RANGARAJAN S? OR RANGARAJAN,
              S? OR ZWILLING M? OR ZWILLING, M?)
S2          2   SONER(2N)TEREK OR AJAY(2N)KALHAN OR NAGAVAMSI(2N)PONNEKANTI
              OR SRIKUMAR(2N)RANGARAJAN OR (MIKE OR MICHAEL)(2N)ZWILLING
S3     507848   SERIALIZ? OR SERIALIS? OR FRAGMENT?
S4          2   S1:S2 AND S3
S5          1   RD (unique items)
? show files
File   9:Business & Industry(R) Jul/1994-2005/May 11
         (c) 2005  The Gale Group
File  13:BAMP 2005/May W1
         (c) 2005  The Gale Group
File  15:ABI/Inform(R) 1971-2005/May 10
         (c) 2005 ProQuest Info&Learning
File  16:Gale Group PROMT(R) 1990-2005/May 11
         (c) 2005 The Gale Group
File  20:Dialog Global Reporter 1997-2005/May 12
         (c) 2005 The Dialog Corp.
File  47:Gale Group Magazine DB(TM) 1959-2005/May 12
         (c) 2005 The Gale group
File  75:TGG Management Contents(R) 86-2005/May W1
         (c) 2005 The Gale Group
File  88:Gale Group Business A.R.T.S. 1976-2005/May 11
         (c) 2005 The Gale Group
File  98:General Sci Abs/Full-Text 1984-2004/Dec
         (c) 2005 The HW Wilson Co.
File 141:Readers Guide 1983-2005/Dec
         (c) 2005 The HW Wilson Co
File 148:Gale Group Trade & Industry DB 1976-2005/May 12
         (c)2005 The Gale Group
File 160:Gale Group PROMT(R) 1972-1989
         (c) 1999 The Gale Group
File 239:Mathsci 1940-2005/Jun
         (c) 2005 American Mathematical Society
File 275:Gale Group Computer DB(TM) 1983-2005/May 12
         (c) 2005 The Gale Group
File 369:New Scientist 1994-2005/Apr W1
         (c) 2005 Reed Business Information Ltd.
File 370:Science 1996-1999/Jul W3
         (c) 1999 AAAS
File 484:Periodical Abs Plustext 1986-2005/May W2
         (c) 2005 ProQuest
File 553:Wilson Bus. Abs. FullText 1982-2004/Dec
         (c) 2005 The HW Wilson Co
File 610:Business Wire 1999-2005/May 11
         (c) 2005 Business Wire.
File 613:PR Newswire 1999-2005/May 12
         (c) 2005 PR Newswire Association Inc
File 621:Gale Group New Prod.Annou.(R) 1985-2005/May 12
         (c) 2005 The Gale Group
File 624:McGraw-Hill Publications 1985-2005/May 11
         (c) 2005 McGraw-Hill Co. Inc
File 634:San Jose Mercury  Jun 1985-2005/May 11
         (c) 2005 San Jose Mercury News
File 635:Business Dateline(R) 1985-2005/May 11
         (c) 2005 ProQuest Info&Learning
File 636:Gale Group Newsletter DB(TM) 1987-2005/May 12
         (c) 2005 The Gale Group
File 647:CMP  Computer Fulltext 1988-2005/Apr W4
```

(c) 2005 CMP Media, LLC
File 674:Computer News Fulltext 1989-2005/May W2
(c) 2005 IDG Communications
File 696:DIALOG Telecom. Newsletters 1995-2005/May 11
(c) 2005 The Dialog Corp.
File 810:Business Wire 1986-1999/Feb 28
(c) 1999 Business Wire
File 813:PR Newswire 1987-1999/Apr 30
(c) 1999 PR Newswire Association Inc
?

EIC #10821687_3

Database Search Request Confirmation

Thank you, GWEN LIANG. Your request (shown below) has been successfully sent to the STIC staff.

Your name: **GWEN LIANG**
Email address: **gwen.liang@uspto.gov**
Employee number: **79180**
Art Unit: **2162**
Office Location: **RND 3B-11**
Phone Number: **x24038**
Maibox Number :

Case serial number: **10821687 (PCT 50424539)**
Class / Subclass(es): **707**
Earliest Priority Filing Date: **4/9/2004**
Format preferred for results: **Paper**

Search Topic Information:
Background: (CON pages 1-5)

Concept of invention: (CON pages 5-6)

Claims: 38, 49, 55 (CLM pages); Support for claims: (CLM-Sup pages)

Drawing: (DRAW pages) * Assignee: Microsoft Corporation

Special Instructions and Other Comments:
*** Preferred searcher: Geoffrey St. Leger**

- **The above referenced pages will be separately submitted to EIC by the examiner to match with this search request.**
- **Search hint:**
- **(must) serialization, header, payload, type field, length field**
- **(combination1) (binary fragment type) (Fig. 1)**
- **(combination2) (Large Object (LOB)) with (type) (Fig. 11(B))**
- **(combination3) (collection or nested or complex) with (object or type) (Fig. 12)**

SYSTEMS AND METHODS FOR FRAGMENT-BASED SERIALIZATION

COPYRIGHT NOTICE AND PERMISSION

FIELD OF THE INVENTION

[0002] This invention relates to computing, and more particularly to storage and transmission of data objects.

Background

BACKGROUND OF THE INVENTION

[0003] Serialization can be defined as the process of storing the state of an object instance to a storage medium. During this process, the public and private fields of an object and the name of the class, are converted to a stream of bytes, which is then written to a data stream. When an object is subsequently deserialized, an exact clone of the original object may be created.

[0004] Consider an object in active computer memory, for example, an object with data describing a person. The person object has a number of subcomponent members, such as name, address, social security number, phone numbers, spouse, height and weight. While the person's

CONC'1/6)

name may be important for a particular application, the height and weight may not be. Thus, the name may remain in active memory where it may be modified, while other fields such as height and weight are evicted from active memory to make room for other data. Ultimately, the person object may no longer be needed by the application, and it may be persisted or transmitted to another computer. To persist or transmit an object, the object must be serialized, which refers to formatting an object in a useful, retrievable way.

[0005] In the example above, the members of an object, such as the person object, are generally uniform for all objects of the same class. Each person object, for example, has the name, address, social security number, phone numbers, spouse, height and weight members. The information changes from person to person, and for some people the information may be unavailable ("null"), but the existence of the same member fields is generally present for all person objects of the person class. As such, a person class may be thought of as the generic person object. A person object is one instance of a person class. This concept of a class and an instance of a class exists in many programming languages. Regardless of the programming language involved, serialization is typically performed on instances of a class, generating serialized objects.

[0006] Objects may comprise members with various types of data. The members may be primitive or complex. Examples of primitive members are "string" such as the name member from the person object, which is a string of letters; and "integer," such as the social security number from the person object, which is an integer. Examples of complex members are "collection," such as the phone numbers member, which comprises more than one primitive—in this case, more than one integer; "nested," which is a member that has some structure beyond a simple primitive member, *e.g.*, the collection of phone numbers, or the spouse member, which refers to another person object; and "subtype," such as a hypothetical "United States address" type that would be a subtype of an address type, and therefore presumably declares additional members such as a U.S. region or U.S. Post Office Box. Members may be described in many different ways, and relate to each other in any number of patterns. Therefore serializing objects such as the person object involves effectively dealing with the various members and the relationships of those members that may be included in the object.

[0007] Serialization of objects presents a number of challenges in the industry. Serialized objects should consume as little storage space as possible. If the size of an object is greatly increased when it is serialized, then the storage cost of the object may be too high. Therefore, compact representation is an important aspect of a serialization format.

CON (2/6)

[0008] Serialized objects should also be efficiently instantiated into active memory. If the processing cost of finding and assimilating the various members of a serialized object is high, it will drain valuable processor resources. Likewise, serialization should allow for instantiation and updating of members of an object without the need to instantiate the entire object. Instantiating the entire person object, for example, only to read or update the person's social security number is a waste of active memory resources needed to store the name, phone number, address, etc. when those members are not involved in the operation.

[0009] Serialization formats should also support all data types that may be contained in an object. A very basic serialization format might only support primitives, but more sophisticated formats should support complex members such as the nested members, collection members, and subtype members described above. While a serialization format should be optimal for objects with few levels of nesting and inheritance, because most objects have this characteristic, it should also support many levels of nesting and inheritance to ensure that the serialization can be flexibly used for a broad range of classes. A serialization format should also be flexible in handling very large members. Some members may be, for example, a music file, a photograph, or a movie, and such large members pose a challenge in serialization that will be explained in greater detail below.

[0010] Previous serialization formats have several notable deficiencies. One such format is known as XML Serialization. XML serialization provides a token for each member. The token comprises metadata that identifies a member, usually a member immediately following the token. Therefore, XML serialization may be visualized as follows:

(token 1) Member 1; (token 2) Member 2; (token 3) Member 3; etc.

[0011] The problems with such a serialization format are, first, verbosity: the storage of metadata tokens with each and every member consumes a large amount of disk space. Second, retrieval is impaired in such a format, because in order to find a desired member, the tokens must be searched. This may involve a high active memory cost, because the most effective way to read or update an object that is serialized in this manner may be to instantiate the entire object.

[0012] Another serialization format is in the "Storage Engine record" format, also referred to as the "SE record," or simply "record" format. This is an a typical database system record format. In this serialization format, members for objects of a given class are stored in uniformly formatted records. Instead of providing metadata that describes each and every member, there is metadata that

CON(3/6)

describes the contents of all the records for objects of a particular class. This can be visualized as provided in Fig. 10.

[0013] The SE record serialization format does not require metadata with each individual member, so it is a more compact serialization technique. Instead, it requires access to metadata describing the layout of the members on disk, such as the *Metadata for Person Objects* table of Fig. 10. A weakness of the SE record format is that it is inflexible in handling members of variable length, such as many of the music files, movies, and images that are stored with objects today. More accurately, flexibility in the SE record serialization comes at a high processing cost. Members of variable length can be stored in such a format, if an offset table is used to identify the locations of variable length data in the record. The consequence of storing an offset table is that whenever a variable length member is updated, the positions of all variable length data that follows it must be adjusted. This can be compared to inserting bytes in the middle of an array --everything to the right of an insert point must be shifted right to make space for inserted new bytes.

[0014] Further , various storage formats have been designed to allow users of databases to efficiently store objects within a database. These storage formats can be better supported with a more flexible serialization format. For example, should be distinguished from the serialization format provided herein. For example United States Patent Application No. 10/692,225, Attorney Docket No. MSFT 2852/306819.01, titled "system and method for object persistence in a database store," is directed to allowing a user to 'import' classes and methods written in an object oriented language like C# into a database. It further allows a user to store C# objects in a database and to invoke methods on the objects. It provides multiple flavors of persistence to a user. A user can define his own serialization format, use Common Language Runtime ("CLR") serialization (provided by C# language itself), or let the SQL server store an object in its own format. These options, particularly the latter, provide a performance advantage, as MICROSOFT SQL SERVER® can retrieve or update some fields of an object without actually instantiating a C# object. Of course, some operations, such as method invocation, still require instantiation of a C# object.

[0015] Similar background and related technology descriptions may be found in United States Patent Application No. 10/692,227, Attorney Docket No. MSFT – 2850/306820.1, titled "System and Method for Storing and Retrieving a Field of a User Defined Type Outside of a Database Store." This application discusses filestreams in UDTs, which may be serialized according to the techniques described herein. Such advanced database technologies can benefit from a more

CON(4/6)

flexible and higher performance serialization format. Likewise, improved techniques for performing operations on serialized objects would better support such advanced database technologies.

[0016] The trade-offs involved in serialization formats are thus metadata on-disk memory overhead of the format, versus active memory overhead of locating a member, versus processing cost of locating a member, versus cost of doing an update, versus flexibility in handling large fields. In light of these trade-offs, there is an ongoing and heretofore unaddressed need in the industry to raise the bar with respect to serialization techniques.

Background

SUMMARY OF THE INVENTION

Concept of invention

[0017] A method and system for fragment-based serialization places one or more members in fragments. Fragments may comprise a header and a payload. A header can provide useful information about the fragment, such as an indication of fragment type and an indication of fragment length. A payload may comprise one or more members of an object. Various fragment types are provided for efficiency and flexibility in storing and retrieving object members. Primitive members may be stored in a fragment with a record format payload. This configuration allows for fast location and updating of primitives. Large Object ("LOB") members may be stored in fragments that have a field for setting forth location types for locations of LOB and FS members. Collections may be stored in a series of fragments, a first fragment to indicate a start of a collection, one or more second fragments to serialize collection elements, and a terminator fragment to indicate the end of a collection. These and other fragment types may be organized according to rules that govern generating fragments, placing members in fragments, and sequencing fragments in a manner that provides additional functionality to the serialization format.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] **Figure 1** is a conceptual illustration of the various fragments which may be used to serialize object members. It shows a Binary Fragment with a payload comprising primitive members in record format, a fragment with a non-record format payload, and a fragment with no payload.

[0019] **Figure 2** presents a fragment with a detailed view of the fragment header. The header shows a selection of potential fields for use in fragment headers, and many fragment headers may omit some of the fields shown.

CON(5/6)

ABSTRACT

A method and system for fragment-based serialization places one or more object members in fragments. Fragments may comprise a header and a payload. A header can provide useful information about the fragment, such as an indication of fragment type and an indication of fragment length. A payload may comprise one or more members of an object. Primitive members may be stored in a Binary Fragment with a record format payload. LOB and FS members may be stored in fragments that have a Value Type field for setting forth additional properties of the fragment. Collections may be stored in a series of fragments, a first fragment to indicate a start of a collection, one or more second fragments to serialize collection elements, and a Terminator Fragment to indicate the end of a collection. Fragment-serialized objects minimize storage overhead while providing fast instantiation and low-cost location and updating.

Concept of invention

CON(6/6)

This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1 – 37. (Canceled)

38. (New) A computer readable medium bearing a computer readable representation of an object that is serialized for efficient retrieval by computer hardware, the computer readable representation comprising:

at least one binary fragment comprising a binary fragment header and a binary fragment payload;

wherein the binary fragment header comprises a type field and a length field;

wherein the type field indicates the fragment is a binary fragment;

wherein the length field indicates a length of the binary fragment payload;

wherein the payload comprises a plurality of primitive data members in storage engine record format; and

wherein said plurality of primitive data members are all of the primitive data members of the object.

39. (New) The computer readable medium of claim 38, wherein the type field indicates that the binary fragment is the only fragment of the object.

40. (New) The computer readable medium of claim 38, further comprising:

at least one Large Object (LOB) fragment comprising a LOB fragment header and a LOB fragment payload;

wherein the LOB header comprises a LOB type field, a value type field, and a LOB length field;

wherein the LOB type field indicates the LOB fragment is a LOB fragment;

wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location;

wherein the LOB length field indicates the a length of the LOB fragment payload.

CLM (1/4)

wherein the collection element type field indicates the collection element fragment is a collection element fragment;

wherein the collection element length field indicates the a length of the collection element payload.

47. (New) The computer readable medium of claim 46, wherein the collection element payload comprises a data member in a collection of data members corresponding to said collection start fragment.

48. (New) The computer readable medium of claim 46, wherein the collection element header further comprises a collection element locator field that provides a unique location of a data member in a collection of data members.

49. (New) A computer readable medium bearing a computer readable representation of an object that is serialized for efficient retrieval by computer hardware, the computer readable representation comprising:

at least one Large Object (LOB) fragment comprising a LOB fragment header and a LOB fragment payload;

wherein the LOB header comprises a LOB type field, a value type field, and a LOB length field;

wherein the LOB type field indicates the LOB fragment is a LOB fragment;

wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location;

wherein the LOB length field indicates the a length of the LOB fragment payload.

50. (New) The computer readable medium of claim 49, wherein the LOB fragment payload comprises a LOB.

CLM(2/4)

51. (New) The computer readable medium of claim 49, wherein the LOB fragment payload comprises a pointer to a LOB location.

52. (New) The computer readable medium of claim 49, wherein the value type field indicates whether the LOB fragment payload comprises an inline LOB, a pointer to a LOB location, or a cell reference.

53. (New) The computer readable medium of claim 49, further comprising:

a collection start fragment comprising a collection start header;

wherein the collection start header comprises a collection start type field and a bit field;

wherein the collection start type field indicates the collection start fragment is a collection start fragment;

wherein the bit field indicates whether a plurality of collection element fragments are ordered or unordered.

54. (New) The computer readable medium of claim 53, further comprising:

a collection element fragment comprising a collection element header and collection element payload;

wherein the collection element header comprises a collection element type field and a collection element length field;

wherein the collection element type field indicates the collection element fragment is a collection element fragment;

wherein the collection element length field indicates the a length of the collection element payload.

55. (New) A computer readable medium bearing a computer readable representation of an object that is serialized for efficient retrieval by computer hardware, the computer readable representation comprising:

a collection start fragment comprising a collection start header;

CLM (3/4)

wherein the collection start header comprises a collection start type field and a bit field;

wherein the collection start type field indicates the collection start fragment is a collection start fragment;

wherein the bit field indicates whether a plurality of collection element fragments are ordered or unordered;

at least one collection element fragment comprising a collection element header and collection element payload;

wherein the collection element header comprises a collection element type field and a collection element length field;

wherein the collection element type field indicates the collection element fragment is a collection element fragment;

wherein the collection element length field indicates the a length of the collection element payload.

56. (New) The computer readable medium of claim 55, wherein the collection element payload comprises a data member in a collection of data members corresponding to said collection start fragment.

57. (New) The computer readable medium of claim 55, wherein the collection element header further comprises a collection element locator field that provides a unique location of a data member in a collection of data members.

CLM (4/4)

REMARKS

Original claims 1-37 have been canceled, and new claims 38-57 are submitted herewith. Claims 38, 49, and 55 are the new independent claims. Minor amendments to the specification and Drawings (Fig. 2) are also submitted herewith. No new matter was added.

In the Official Action, dated June 28, 2005, certain phrases in the specification were objected to. Claims 7, 9, 12, 13, 20 and 22 were rejected under 35 U.S.C. § 112, second paragraph. Claims 1-13, 14-25, and 27 were rejected under 35 U.S.C § 101. Claims 1- 10, 14 – 23, 26 and 27 were further rejected under 35 U.S.C § 103(a) as allegedly obvious over alleged applicant admission ("Admission") in view of U.S. Pat. No. 5,634,123 ("Bennion"). Claims 11, 12, 24 and 25 were further rejected under 35 U.S.C § 103(a) as allegedly obvious over alleged Admission in view of Bennion and further in view of U.S. Pat. No. 5,568,639 ("Wilcox").

Rejections Under 35 U.S.C. § 112, Second Paragraph, 35 U.S.C. § 101, and 35 U.S.C. § 103

Because the original claims have been canceled, the above rejections under 35 U.S.C. § 112, § 101, and § 103(a) are moot. New claims 38 – 57 define over Bennion and Wilcox as will be apparent.

For example, new claims 38, 49, and 55 require that fragments be identified in a type field as a binary fragment, a LOB fragment, or a collection start and collection element fragment, respectively. The fragments are thus identified as having the particular properties set forth in the claims. A system reading these type fields will be configured to react to the type field to leverage the unique properties of corresponding fragments. Neither Bennion nor Wilcox disclose fragment types with the unique properties of the invention.

↓ general claim support

↑

CLM-Sup (1/2)

With respect to the binary fragment in claim 38, by definition this comprises a payload with a plurality of primitive data members in storage engine record format. Moreover, it is identified in a type field as a fragment that will contain such a payload. While Wilcox discloses a "binary type" (col. 15, lines 46-52), it is defined therein as "used to disclose data that is anonymous to the typing system of the invention." Furthermore, claim 38 requires that the binary fragment contain all primitive members of a particular object. LOB and FS members are not considered to be primitive members.

With respect to the LOB fragment in claim 49, it has a value type field that "indicates whether the LOB fragment payload comprises an inline LOB or a pointer to a LOB location." Neither Bennion nor Wilcox disclose such a value type field. Note that element 204 in Bennion's Fig. 2 is described as "a one-byte field that specifies an attribute type that is used by the application program106 in interpreting the stored data." This is quite different from the the value type field limitation in claim 49.

Finally, with respect to the collection start and collection element fragments in claim 55, applicants point out that the collection start fragment contains a bit field that "indicates whether a plurality of collection element fragments are ordered or unordered." This limitation cannot be found in Bennion, Wilcox, or any other reference of record.

Objections to the Specification

Applicants have addressed the outstanding objections by amending the specification as set forth above. Paragraph **0012** has been amended by removing the word "an" as suggested by

CLM-Sup(2/2)

flexible and higher performance serialization format. Likewise, improved techniques for performing operations on serialized objects would better support such advanced database technologies.

[0016] The trade-offs involved in serialization formats are thus metadata on-disk memory overhead of the format, versus active memory overhead of locating a member, versus processing cost of locating a member, versus cost of doing an update, versus flexibility in handling large fields. In light of these trade-offs, there is an ongoing and heretofore unaddressed need in the industry to raise the bar with respect to serialization techniques.

SUMMARY OF THE INVENTION

[0017] A method and system for fragment-based serialization places one or more members in fragments. Fragments may comprise a header and a payload. A header can provide useful information about the fragment, such as an indication of fragment type and an indication of fragment length. A payload may comprise one or more members of an object. Various fragment types are provided for efficiency and flexibility in storing and retrieving object members. Primitive members may be stored in a fragment with a record format payload. This configuration allows for fast location and updating of primitives. Large Object ("LOB") members may be stored in fragments that have a field for setting forth location types for locations of LOB and FS members. Collections may be stored in a series of fragments, a first fragment to indicate a start of a collection, one or more second fragments to serialize collection elements, and a terminator fragment to indicate the end of a collection. These and other fragment types may be organized according to rules that govern generating fragments, placing members in fragments, and sequencing fragments in a manner that provides additional functionality to the serialization format.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] **Figure 1** is a conceptual illustration of the various fragments which may be used to serialize object members. It shows a Binary Fragment with a payload comprising primitive members in record format, a fragment with a non-record format payload, and a fragment with no payload.

[0019] **Figure 2** presents a fragment with a detailed view of the fragment header. The header shows a selection of potential fields for use in fragment headers, and many fragment headers may omit some of the fields shown.

DRAW (1/5)

[0020] **Figure 3** presents several exemplary object classes for which fragment sequences are provided in the description, in accordance with various embodiments of the invention.

[0021] **Figure 4** is a flowchart demonstrating steps for generating fragments for the primitive members of an object when there are no nested members in the object.

[0022] **Figure 5** is a flowchart demonstrating steps for generating fragments for the primitive members of an object when there are nested members in the object.

[0023] **Figure 6** is a flowchart demonstrating steps for generating fragments for collection members of an object.

[0024] **Figure 7** is a flowchart demonstrating steps for generating fragments for the LOB and FS members of an object.

[0025] **Figure 8** is a flowchart demonstrating steps for the process of placing an entire object, with members various types, into fragments.

[0026] **Figure 9** illustrates objects that have been serialized in accordance with various embodiments of the invention as they may be stored in a single column of a database.

[0027] **Figure 10** illustrates a prior art record serialization format in which metadata is provided for all records, and corresponding data conforms to the format specified in the metadata.

[0028] **Figures 11(A-H)** illustrate various fragment types for use in serializing data in accordance with preferred embodiments of the invention.

[0029] **Figure 12** is a top-level diagram of a fragment sequence for the *tPartTimeEmployee* object displayed in Fig. 3. This fragment sequence may contain additional fragments for each level of nesting.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS.

[0030] Certain specific details are set forth in the following description and figures to provide a thorough understanding of various embodiments of the invention. Certain well-known details often associated with computing technology are not set forth in the following disclosure, however, to avoid unnecessarily obscuring the various embodiments of the invention. Further, those of ordinary skill in the relevant art will understand that they can practice other embodiments of the invention without one or more of the details described below. Finally, while various methods are described with reference to steps and sequences in the following disclosure, the description as such is for providing a clear implementation of embodiments of the invention, and the steps and sequences of steps should not be taken as required to practice this invention.

DRAW (2/5)

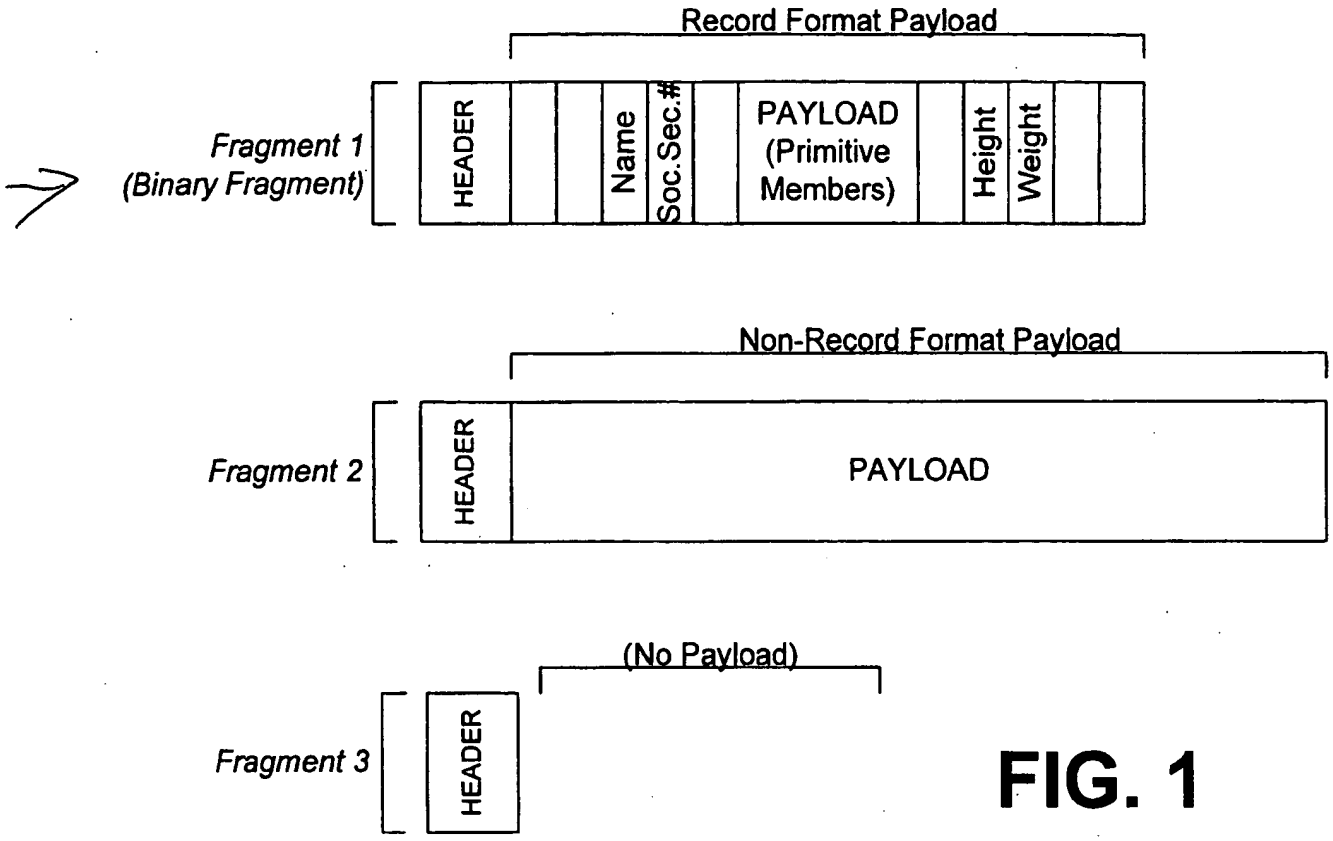


FIG. 1

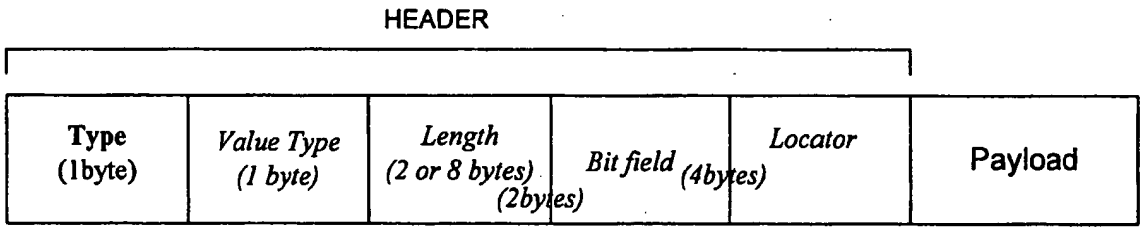


FIG. 2

DRAW(3/L-1)

Metadata for Person Objects

Name	Phone #	Address	SS#	Weight

Person Object Records

Henry	547-5268	2 N. 53rd St.	383-99-9876	230lbs
Margaret	549-9254	4 N. 67th St.	398-74-4565	160lbs

(Prior Art)

FIG. 10

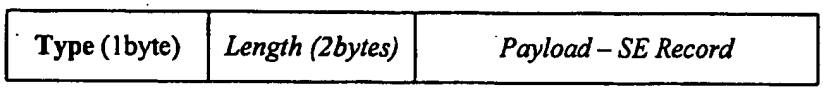


FIG. 11(A)

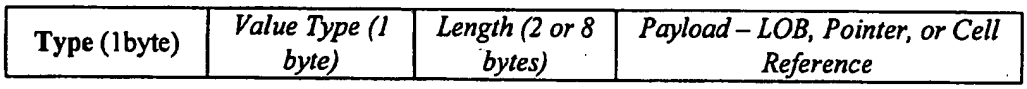


FIG. 11(B)

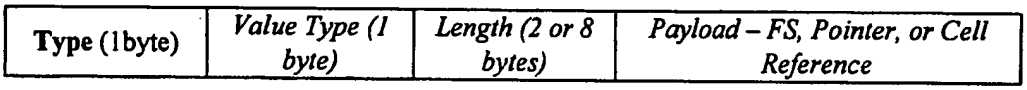


FIG. 11(C)

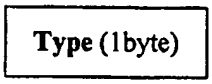


FIG. 11(D)

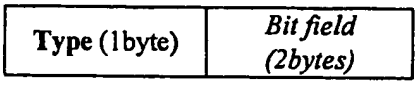


FIG. 11(E)

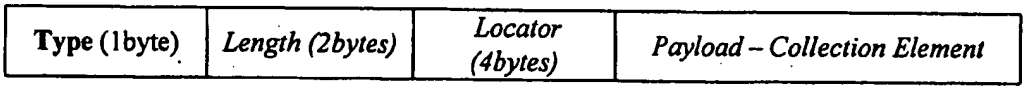


FIG. 11(F)

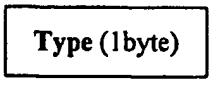


FIG. 11(G)

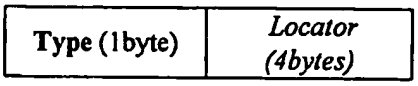


FIG. 11(H)

DRAW(4/5)

collection or nested or complex

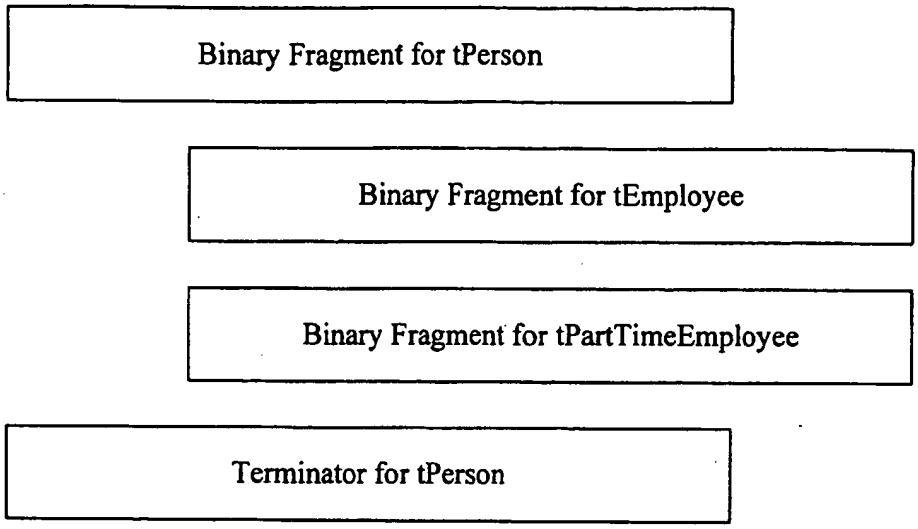


FIG. 12

DRAW (5/5)